Dozens of New Products For Atari ST And Amiga

COMPUTE!

$3.00
August
1986
Issue 75
Vol. 8, No. 8
$4.25 Canada
02193
ISSN 0194-357X

The Leading Magazine Of Home, Educational, And Recreational Computing

The Commodore 64C
Report From The Summer
Consumer Electronics Show

Tightrope
A Thrilling Action Game
For Commodore 64, 128,
Atari, Apple, Amiga,
IBM PC/PCjr

Sprite 32
For Commodore 64
Dozens Of Sprites
Onscreen At Once

MODified Shapes
For Atari ST
Colorful Graphics
In ST BASIC

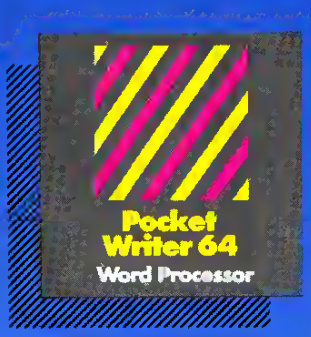
A New IF/THEN
For Atari BASIC

Password Protection
For Apple II Disks

Batch Files
With IBM BASIC

0 71486 02193 08

Solutions!



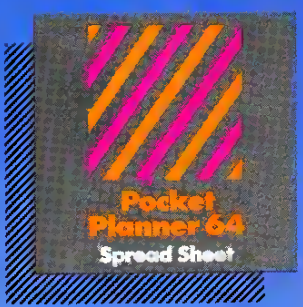
PW 128/64 Dictionary
also available at $14.95 (U.S.)

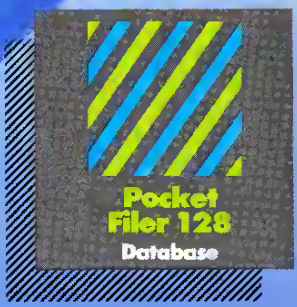
MAIL ORDERS

CRYSTAL COMPUTER INC.
In Michigan 1-517-224-7667
Outside Michigan 1-800-245-7316

CANADIAN DEALER ENQUIRIES

INGRAM CANADA LTD.
1-416-738-1700





Only The Name Is New

The professional, full-featured software line from Digital Solutians is now called Pocket Software.
Pocket Writer 128/64.
Pocket Filer 128/64.
Pocket Planner 128/64.
The names are new, but this super software is still the same.

From now on, when you hear the word Pocket, it means software that's full-featured, handy and easy to use.

Pocket Software at prices that won't pick your pocket.

Best-selling software for Your Commodore 128 or 64

You want the very best software you can find for your Commodore 128 ar 64, right?

Yau want integrated saftware — word processing, database and spreadsheet applications — at a sensible price. But, you also want tap-of-the-line features. Well, aur Packet 128/64 software gaes ane better.

With Packet 128 ar 64, yau'll find all the features you can imagine . . . and then some. And Packet 128/64 is so easy ta use, you won't even need the reference guide. On-screen and in memary instructions will have yau up and running in less than 30 minutes, even if you've never used a computer before.

The price? It's as low as you'd expect for a line of software called 'Pocket'. Suggested Retail Price for the 64 software is $39.95 (U.S.) and $49.95 (U.S.) for the 128. Any of the 64 products may be upgraded to their 128 version for $15.00 (U.S.) + $3.00 shipping and handling. (Available ta registered owners from Digital Solutions Inc. only.)

Pocket Writer 128 or 64, Pocket Planner 128 or 64 and Pocket Filer 128 or 64 . . . **Solutions** at sensible prices from Digital Solutions Inc.

International & Distributor enquiries to:

Serious software that's simple to use.

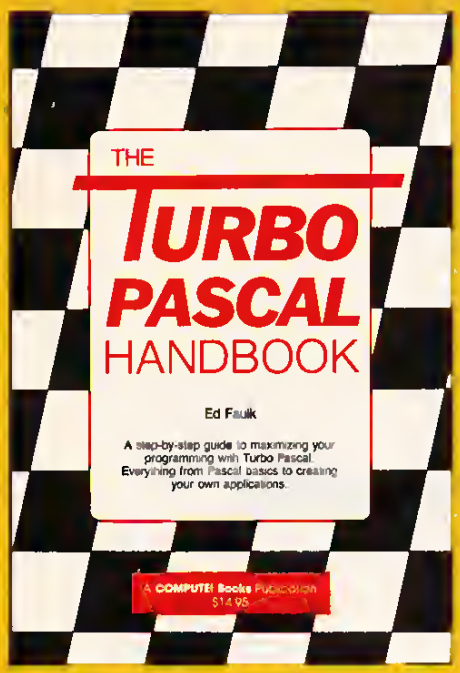
Digital Solutions Inc.

30 Wertheim Court, Unit 2
Richmand Hill, Ontario
Canada L4B 1B9
telephone (416) 731-8775

Pocket Writer 128 and 64 are naw available in French.

Explore Pascal with

THE TURBO PASCAL HANDBOOK from COMPUTE!.



The Turbo Pascal Handbook
Edward P. Faulk
With *The Turbo Pascal Handbook* and *Turbo Pascal* from Borland International, you'll be gently guided, step-by-step, until you're creating your own powerful applications in this impressive computer language.
$14.95 ISBN 0-87455-037-8

This information-packed book from COMPUTE! is an outstanding resource and programming guide. And it's written in COMPUTE!'s bestselling style so that even beginning programmers can quickly and easily understand all the applications.

Ask for *The Turbo Pascal Handbook* at your local computer store or bookstore. Or order directly from COMPUTE!. Call toll free 1-800-346-6767 (in NY 212-887-8525) or mail the attached coupon with your payment (plus $2.00 shipping and handling per book) to COMPUTE! Books, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

Note: You'll need *Turbo Pascal* in order to use this book. The software is not included with *The Turbo Pascal Handbook*.

Yes! Send me _____ copies of *The Turbo Pascal Handbook* at $14.95 each.
My payment is enclosed.

ALL ORDERS MUST BE PREPAID IN U.S. FUNDS

Subtotal _____
NC residents add 4.5% sales tax _____
Shipping and handling _____
($2.00 per book in U.S. and surface mail, $5.00 per book airmail.)
Total enclosed _____

☐ Payment enclosed (check or money order)
☐ Charge ☐ Visa ☐ MasterCard ☐ American Express

Account No. _____ Exp. Date _____
(Required)

Name _____

Address _____

City _____ State _____ Zip _____

Please allow 4–6 weeks for delivery.

36603711

COMPUTE!™ Publications, Inc. abc
One of the ABC Publishing Companies
825 7th Avenue, 6th Floor, New York NY 10019
Publishers of COMPUTE! COMPUTE!'s Gazette COMPUTE!'s Gazette Disk COMPUTE! Books and COMPUTE!'s Apple Applications

COMPUTER MAIL ORDER

THE CMO ADVANTAGE

TO ORDER
CALL TOLL FREE
1-800-233-8950
DEPARTMENT A208
TELEX 5106017898
OR MAIL YOUR ORDER TO:
COMPUTER MAIL ORDER
Department A208
477 E. Third Street
Williamsport, PA 17701

VISA · MasterCard · DINERS CLUB

POLICY
Add 3% (Minimum $7 00) shipping and handling Larger shipments may require additional charges Personal and company checks require 3 weeks to clear For faster delivery use your credit card or send cashier's check or bank money order Pennsylvania residents add 6% sales tax All prices are subject to change and all items are subject to availability Defective software will be replaced with the same item only Hardware will be repaired or replaced at our discretion within the the terms and limits of the manufacturer's warranty We cannot guarantee compatibility All sales are final and returned shipments are subject to a restocking fee

EDUCATIONAL INSTITUTIONS
CALL TOLL FREE
1-800-221-4283
CUSTOMER SERVICE
& TECHNICAL SUPPORT
1-717-327-1450

CANADIAN ORDERS
1-800-268-3974
Ontario/Quebec
1-416-828-0866
In Toronto
1-800-268-4559
Other Provinces
TELEX: 06-218960
2505 Dunwin Drive,
Mississauga, Ontario
Canada L5L1T1

All prices shown are for U S A orders Call the Canadian Office for Canadian prices

THE CMO ADVANTAGE
- Next day shipping on all in-stock items
- Free easy access order inquiry
- Orders from outside Pennsylvania save state sales tax
- Free technical support from our factory trained technicians
- There is no limit and no deposit on C O D orders
- There is no extra charge for using your Visa or MasterCard and your card is not charged until we ship
- No waiting period for cashier's checks
- We accept purchase orders from qualified corporations Subject to approval
- Educational discounts available to qualified institutions (See the toll free educational phone number above)
- FREE CATALOG MEMBERSHIP

HOME COMPUTERS

ATARI
65XE (94K).....$99.99
130XE (129K).....$139.00
520ST (512K).....$369.00
5205T Monochrome Syetem
- 520ST with modulator
- disk drive
- mouse
- logo
- Basic
- 1st Word
- monochrome monitor

LOW, LOW SYSTEM PRICE **$649.00**

520ST Color System
- 520ST with moduletor
- disk drive
- mouse
- logo
- Basic
- 1st Word
- color monitor

LOW, LOW SYSTEM PRICE **$769.00**

800XL 64K	$69 99
1010 Recorder	$49 99
1050 Disk Drive	$149 00
1020 Printer	$29 99
1027 Letter Quality Printer	$129 00
1030 Direct Connect Modem	$59 99
Comrex 220 Atan	$89 99

APPLE
APPLE IIe	CALL
APPLE IIc	CALL
IIc LCD Display	$329 00

COMMODORE
Amige Peckege
- 512K • 1 Drive
- RGB Monitor **$1399.00**

C64 Peckege
- C64 • C1541
- Taxan 220 **$499.00**

C129 Peckege
- C128 • C1571
- NAP8562 Monitor **$779.00**

C129 Computer.....$299.00
C1571 (Disk Drive for C129).....$249.00
C1902 (RGB 13" Monitor for C129.....CALL
C1970 (Modem for C129)$179.00

C1530 Datasette	$39 99
C1660 Auto Modem	$59 99
DPS 1101 Daisy Printer	$339 00
Comrex 220 (C64 Interface)	$89 99
Xetec SuperGraphix 8K	$69 99
Micro R&D 128/84 Interface	$59 99
C128 Universal Monitor Cable	$19 99

PORTABLE COMPUTERS

NEC
PC-8401 LS	$699.00
PC-8201 Portable Computer	$339 00
PC-8231 Disk Drive	$599.00
PC-8221A Thermal Printers	$149 00
PC-8281A Data Recorder	$99 99
PC-8201-06 8K RAM	$59 99

SHARP
PC-1350	$149 00
PC-1261	$149 00
PC-1500A	$169 00
PC-1250A	$89 99
CE-125 Printer/Cassette	$129 00
CE-150 Color Printer Cassette	$149 00
CE-161 16K RAM	$129 00

HEWLETT PACKARD
41CV	$139.00
41CX	$199 00
HP 11C	$49 99
HP 12C	$75 99
HP 15C	$75 99
HP 16C	$89 99
HPIL Module	$98 99
HPIL Cassette or Printer	$359 99
Card Reader	$143 99
Extended Function Module	$63 99
Time Module	$63 99

We stock the full line of HP calculator products

MODEMS

ANCHOR
Volksmodem	$59 99
Volksmodem 300/1200	$139 00
Signalman Express	$209.00
Lightning 2400 Baud	$329 00
Expressi (PC Halfcard)	$149 00
6480 (64/128) 300/1200 Baud	$139 00
520 1200 BPS	$149 00
Lightening Helf Cerd	$349 00

AST
Reach 1200 Baud Half Card $399 00

DIGITAL DEVICES
AT300 - 300 Baud (Atari) $99 99

EVEREX
1200 Baud Internal (IBM/PC) $179 00

Hayes
Smartmodem 300	$139.00
Smartmodem 1200	$389 00
Smartmodem 1200B	$359.00
Smartmodem 2400	$599.00
Micromodem IIe	$149.00
Smart Com II.	$89 99
Chronograph	$199.00
Trenset 1000	$309 00

Novation
Smert Cat Plus	$299 00
J-Cat	$99 99
Novation 2400	$499 00
Apple Cat II.	$219 00
212 Apple Cet II	$379 00
Apple Cat 212 Upgrade	$229 00

QUADRAM
Quadmodem II
300/1200	$339 00
300/1200/2400	$499 00

SUPRA
MPP-1064 AD/AA (C-64) $69 99

ACCESSORIES

AMARAY
80 Column Printer Stand $14 99

CURTIS
Side Mount SS-1	$19 99
Side Mount AT SS-2	$34 99
Universal Stand SS-3	$19 99
Diamond SP-1	$29 99
Emerald SP-2	$39 99
Sapphire SPF-1	$49 99
Ruby SPF-2	$59 99
Universal Printer Stend	$14 99
Static Mat	$29 99

DATA SHIELD
300 Watt Backup	$379 00
500 Watt Backup	$589.00
Turbo 350 Watt Backup	$449 00
P125 Power Director	$99 99
P150 Power Director w/Modem	$119 00

KENSINGTON
Master Piece	$99 99
Master Piece +	$119 00

KEYTRONICS
KB5150/KB5151/KB5151Jr	CALL
KB5152B/KB5153/KB5149Jr	CALL

MEMORY CHIPS
4164 RAM Chips	(ea) $1 99
128 RAM Chips	(ea) $12 99
256 RAM Chips	(ea) $10 99

Polaroid
Palette	$1399 00
Power Processor	$229 00
Illuminated Slide Mounter	$39 99
Polacolor 2 Pack film	$18 99

DISKETTES

GENERIC
DS/DD w/Flip'n'File 10 $11 99

IBM
5¼" OS/DD floppy disks
(Box of 10). $26 99

maxell
3½" SS/DD (10)	$18 99
3½" 5 pack SS/DD/Case	$9 99
3½" DS/DD (10)	$29 99
5¼" MD-1 SS/SD (10)	$11 99
5¼" MD-2 DS/DD (10)	$16 99
5¼" MD-2-HD for AT (10)	$29 99

SONY
MD1-5¼" SS/SD (10)	$9.99
MD2-5¼" DS/DD (10)	$13 99
MD2HD-5¼" High Density (10)	$28 99
MFD1-3½" SS/DD (10)	$18 99
MFD2-3½" DS/DD (10)	$29 99

Verbatim.
5¼" SS/DD	$12 99
5¼" DS/DD	$24 99
Disk Analyzer	$24 99

DISK HOLDERS
AMARAY
50 Disk Tub 5¼"	$9 99
30 Disk Tub 3½"	$9 99
100 Disk Tub 5¼" w/lock	$19 99

INNOVATIVE CONCEPTS
Flip'n File 10	$2 49
Flip'n File 50	$14 99
Flip'n File 50 w/lock	$19 99
Flip'n File Data Case	$9 99

DRIVES

IOMEGA
A110H Single 10	CALL
A210H 10 + 10	CALL
A120H Single 20	CALL
A220H 20 + 20	CALL
Save on 10 & 20 Carts	CALL

TALLGRASS
25, 35, 50, 80 meg (PC) from $1299.00

IRWIN
Tape Backup CALL

EVEREX
60 Meg Internal Backup System	$799.00
20 Meg Streamer	$699.00

CORE
AT20-AT72MB CALL

MOUNTAIN
Hard Card 20 CALL

PRIAM
40, 60 MB Inner Space	CALL
Shared Data	CALL
Shared Space	CALL

RACORE
1500 Expension Chassis	$379.00
1500 Chassis w/DMA	$429.00
2101 256K Memory	$179.00
2103 512K Memory	$199.00

CMS
10 Meg with controller	$399.00
20 Meg with controller	$499.00
Rambo-AT	$229.00

ALLIED TECHNOLOGY
Apple II,it +,IIe ½ height $109.00

INDUS
Atari GT	$199.00
C-64 /128 GT	$199.00

MSD
SD1 C-64 Sirigle	$219.00
SD2 C-64 Dual	$469.00

TANDON
320K 5¼" (PC) $119.00

TEAC
320K 5¼" $119.00

DEPT. A208 CALL TOLL-FREE

SOFTWARE FOR IBM

ANSA SOFTWARE
Peradox $499.00

ASNTON-TATE
Fremework II $399.00
dBaee III Plus $399.00

SATTERIES INCLUOEO
Isgur Portfolio $159.00

BORLANO
Lightening $59.99
Sidekick (unprotected) $57.99
Reflex $59.99
Trevelling Sidekick $44.99
Turbo Prolog $64.99

CENTRAL POINT
Copy II PC-Backup $29.99
PC Qption Board $84.99

OECISION RESOURCES
Chartmaster $229.00
Signmaster $159.00
Diagram Master $209.00
Map Master $259.00

FIFTN OENERATION
Fast Beck 99.99

FUNK SOFTWARE
Sideways $44.99

NARVARD SOFTWARE INC.
Total Project Manager $269.00
Presentation Graphics $239.00

LIFETREE
Volkswriter III $159.00

LIVINO VIOEOTEXT
Think Tenk $109.00
Reedy $64.99

LOTUS
Symphony CALL
1-2-3 Version 2 CALL

MECA SOFTWARE
Managing Your Money 2.0 $99.99
Manage Your Market $89.99

MICROPRO
Easy $94.99
WordStar 2000 $239.00
WordStar 2000+ $289.00
WordStar Pro w/GL Demo $189.00

MICRORIM SOFTWARE
R Base 4000 $249.00
R Base 5000 $359.00
Clout 2.0 $129.00

MICROSOFT
Flight Simulator $34.99
MultiPlan $129.00
Word 3.0 $279.00
Mouse $139.00

MICROSTUF
Crosstalk XVI $89.99
Crosstalk Mark IV $149.00
Remote $89.99

MULTIMATE
Multi Mate Word Proc $219.00
Advantage $269.00
Qn File $89.99
Just Write $89.99

NOUNEMON
Intuit $69.99

NORTON
Norton Utilities 3.1 $57.99

ONE STEP
Golf's Best $34.99

PFS: IBM
Proof $59.99
File/Graph (ea) $84.99
Report $74.99
Write/Proof Combo $84.99

PROFESSIONAL SOFTWARE
Write-N-Spell $89.99

THE SOFTWARE OROUP
Enable $329.00

SATELLITE SYSTEMS
Word Perfect 4.1 $219.00

SORCIM/IUS
Accounting
AP/AR/GL/INV/QE (ea) $299.00
SuperCalc III $199.00
EasyWriter II System $239.00
Super Project $199.00

SUSLOOIC
Jet $37.99

PRINTERS

Canon
A40,A50,A55 CALL
LBP-8A1 Laser CALL

CITIZEN
MSP-10 (80 col.) $279.00
MSP-15 (132 col.) $389.00
MSP-20 (80 col.) $349.00
MSP-25 (132 col.) $509.00

C.ITOH
Prowriter 1550P $349.00
Starwriter 10-30 $399.00
3500 Tri Printer $1499.00

corona
Lazer LP-300 $2799.00

DIABLO
D25 Daisywheel $549.00
635 Daisywheel $1099.00
D80IF Daisywheel CALL

daisywriter
2000 $699.00

EPSON
Homewriter 10, LX-80 CALL
FX-85, FX-286, RX-100 CALL
DX-10, DX-20, DX-35 CALL
SQ-2000, Hi-80, HS-80, AP-80 CALL
LQ-800, LQ-1000 CALL

JUKI
6000 Letter Quality CALL
6100 Letter Quality CALL
6200 Letter Quality CALL
6300 Letter Quality CALL
6500 Letter Quality CALL
5510 Dot Matrix Color CALL

LEOEND
808 Dot Matrix 100 cps $179.00
1080 Dot Matrix 100 cps $259.00
1380 Dot Matrix 130 cps $289.00
1385 Dot Matrix 185 cps $339.00

NEC
3000 Series $779.00
8000 Series $1099.00
ELF 360 $399.00
P560, P660 P760 CALL

OKIDATA
182, 183, 192, 193, 2410, 64 CALL
Qkimate 10 (Specify C64/Atari) $189.00
Qkimate 20(IBM), 282, 283 CALL

Panasonic
KX1080 NEW
KX1091 $259.00
KX1092 $389.00
KX1592 $469.00
KX1595 $659.00

QUADRAM
Quedjet $399.00
Quad Laser CALL

SILVER-REED
500 Letter Quality $219.00
550 Letter Quality $299.00
800 Letter Quality $699.00

star
SG-10C (C64 Interface) CALL
SB/SD/NX/SR Series CALL
Powertype Letter Quality CALL

Texas Instruments
TI850 $529.00
TI855 $639.00
TI865.. $799.00

TOSHIBA
P321 (80 column) $499.00
P341 (132 column) $799.00
P351 (132 column) $1049.00

MULTIFUNCTION CARDS

AST
RamVantage $349.00
Rampage-PC $379.00
Rampage-AT CALL
Six Pack Plus $229.00
I/O Plus II $139.00
Advantage-AT $369.00
Preview Mono $299.00
PC Net Cards $379.00
5251/11 On-line $669.00
5251/12 Remote $579.00
5251/11 Plus $719.00

dca
IRMA 3270 $879.00
IRMA Print $999.00
IRMA Smart Alec $779.00

EVEREX
Edge Card $259.00
Graphics Edge $219.00
Magic Card II $159.00
Magic Card I $99.99
EGA Video CALL

HERCULES
Graphics $299.00
Color $159.00

IDEAssociates
IDEA 5251 $549.00

INTEL
PCNC8087 5MHz CALL
PCNC8087-2 8 MHz FOR
PCNC80287 6 MHz YOUR
1010 PC-Above Board PC
1110 PS-Above Board
2010 AT-Above Board

MYLEX
The Chairman $439.00

PARADISE
Color/Mono Card $149.00
Multi Display Card $199.00
Five Pack C, S, 0-384K $99.99
High Res Mono $169.00
EGA Video Card $379.00

PERSYST
Bob Board $359.00

QUAORAM
Quadport-AT $119.00
Liberty-AT (128K) $349.00
The Gold Quadboard $449.00
The Silver Quadboard $239.00
Expanded Quadboard $179.00
Liberty $309.00
QuadSprint $499.00
QuadLink $399.00
QuadColor $199.00
Quadboard-AT $399.00
8600 E G A card $399.00

STS
EGA Plus $379.00

TECMAR
Captain - 64 $199.00
Graphics Master $469.00

VIOEO-7
EGA $399.00

INTERFACES

AST
Multi I/O (Apple II) $149.00

DIGITAL DEVICES
Ape Face (Atari) $49.99
U-Print A (Atari) $54.99
U-A16/Buffer (Atari) $74.99
U-Call Interface (Atari) $39.99
U-Print C (C64) $49.99
P-18 Print Buffer $74.99
U-Print 16 apple IIc $89.99

MICRO R & O
Apple IIc Parallel $49.99
Keypro 2000 Parallel $49.99
C64/128 $59.99

Orange Micro
Grappler CD (C64) $89.99
Grappler Plus (IIe, IIc) $89.99
Grappler C (IIc) $89.99
Grappler 16K (IIe, II+) $139.00

PRACTICAL PERIPHERALS
Graphicard $89.99
Serial Card $99.99
Microbuffer II+ 64K $169.00

QUADRAM
Microfazer from $139.00
Efazer (Epson) from $79.99

IBM

IBM PC SYSTEMS
Configured to your specifications.
Call for Best Price!
ISM-PC, ISM-XT, IBM-AT

AT&T
Safari (7300) CALL
6300 CALL

corona
PPC400 Dual Portable $1289.00
PPCXT 10 meg Portable $1989.00
PC40022 Dual Desktop $1389.00
PC400-HD2 10 meg $1989.00

ITT
ITT X-TRA
256K, 2 Drive System CALL
256K,10 meg Hard Drive System CALL
XP5, 20 meg CALL

KAYPRO
KP-2000 Portable CALL
Kaypro PC CALL

PCTOO **IBM CLONE**
640K MB-256 installed, 150 Watt Power Supply, XT Keyboard, 360K Floppy, 20 Meg Hard Drive **$999.00**

SPERRY
Sperry-AT as low as $1749.00
Sperry-IT as low as $2699.00
Call for Specific Configuration!
All Models CALL

Zenith
PC-138 Series, PC-148 Series, PC-158 Series, PC-160 Series, PC-171 Series, AT-200 Series CALL

MONITORS

AMDEK
Video 300A Amber $129.00
Video 310A Amber TTL $159.00
410 TTL Amber or Green CALL
Color 600 Hi-Res RGB $399.00
Color 722 Dual Mode $529.00
Color 725 CALL
Color 730 CALL

MACNAVOX
515 RGB/Composite $289.00
613 TTL Green $99.99
623 TTL Amber $99.99

NEC
JB1270G/1275A (ea.) $99.99
JB1260G TTL Green $129.00
JB1285A TTL Amber $129.00
JC1401 Multi Sync RGB CALL

PRINCETON
MAX-12 Amber $179.00
HX-9 9" RGB $469.00
HX-9E Enhanced $519.00
HX-12 12" RGB $469.00
HX-12E Enhanced $559.00
SR-12 Hi-Res $599.00
SR-12P Professional $699.00

QUADRAM
8400 Quadchrome I $499.00
8410 Quadchrome II $339.00
8420 Amberchrome $179.00
8500 Quad Screen $1449.00

TAXAN
115 12" Green $119.00
116 12" Amber $129.00
121 TTL Green $139.00
122 TTL Amber $149.00
220 14" Color Composite $179.00
620 640x200 RGB $439.00
630 640x200 RGB $489.00
640 720x400 RGB $539.00

Zenith
ZVM 1220 Amber $99.99
ZVM 1230 Green $99.99
ZVM 1240 IBM Amber $149.00
ZVM 135 RGB $459.00
ZVM 1330 RGB $459.00
ZVM 1360 RGB CALL
ZVM 1380 E G Comp CALL

COMPUTER MAIL ORDER

Introducing Certificate Maker.™

Because accomplishments deserve to be recognized.

Offer Congratulations! Say Thanks! Have Fun!

Giving someone a certificate is a wonderful way to recognize an outstanding achievement. It's also a perfect way to have a little fun.

Certificate Maker™ gives you **over 200 professionally designed certificates.** From strictly official to fun and witty, there's something for everyone and every occasion. So you can surprise a family member, praise a student, applaud an athlete and honor an employee with great looking certificates. And each one will be as personal, professional and special as you choose.

Personalized certificates in minutes.

Simply choose a certificate, select a border, type your message; add a date and signature... then print! It's that quick and that easy.

You can even create a name file and automatically personalize certificates for everyone in your class or club!

Over 200 exciting Certificates, Awards, Diplomas, and Licenses.

SPRINGBOARD

Now available for Apple II+, IIe, IIc, and IBM PC • Springboard Software, Inc • 7808 Creekridge Circle • Minneapolis, MN 55435 • (612) 944-3915

COMPUTE!

AUGUST 1986
VOLUME 8
NUMBER 8
ISSUE 75

FEATURES

REVIEWS

COLUMNS AND DEPARTMENTS

THE JOURNAL

NOTE: See page 121 before typing in programs.

GUIDE TO ARTICLES AND PROGRAMS

AP Apple, **Mac** Macintosh, **AT** Atari, **ST,** Atari ST, **V** VIC-20, **64** Commodore 64, **+4** Commodore Plus/4, **16** Commodore 16, **128** Commodore 128, **P** PET/CBM, **TI** Texas Instruments, **PC** IBM PC, **PCjr** IBM PCjr, **AM** Amiga, *General Interest

TOLL FREE Subscription Order Line
800-247-5470 (In IA 800-532-1272)

COMPUTE! Publications, Inc.
Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies
ABC Publishing, President, Robert G. Burton
1330 Avenue of the Americas, New York, New York 10019

COMPUTE! The Journal for Progressive Computing (USPS. 537250) is published monthly by COMPUTE! Publications, Inc., 825 7th Ave., New York, NY 10019 USA. Phone (212) 265-8360. Editorial Offices are located at 324 West Wendover Avenue, Greensboro, NC 27408. Domestic Subscriptions: 12 issues, $24. POSTMASTER: Send address changes to **COMPUTE!** Magazine, P.O. Box 10955, Des Moines, IA 50950. Second class postage paid at Greensboro, NC 27403 and additional mailing offices.

Entire contents copyright ©1986 by COMPUTE! Publications, Inc. All rights reserved.

ISSN 0194-357X.

Editor's Notes

The recent Summer Consumer Electronics Show was both interesting and disappointing. Last year at this time, the industry was reeling from a tremendous downturn in sales growth, and the resulting shakeout had otherwise stable vendors describing those times as the end of the entire personal computer industry. A year later, we're still here, and the doom and gloom forecasters have retrenched. We're a wiser, more mature, and perhaps more stable industry, and the attitude among the exhibitors at the show was much more upbeat. We heard talk of steadily improving sales, enthusiasm for new products, and a better holiday season on the horizon. We also heard a general level of enthusiasm for the hard-charging Atari Corporation, and a more specific level of disappointment at the Commodore showing. Atari had a large and impressive booth, impressive in that it contained dozens of smaller exhibits where independent vendors demonstrated software for the ST series. Visitors thus immediately encountered a tremendous amount of activity encompassed in a group of highly supportive people.

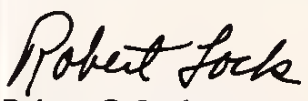
The Commodore appearance evoked a mixture of concern and amazement. Remember, we're talking about a company here with an active, enthusiastic installed base of literally millions of computers. We're talking about a computer series called the 64 that just keeps going, the 128 with a success record that we suspect even impresses Commodore, and the Amiga. One of the most technologically superior computers on the market, the Amiga continues to suffer at the hands of the superior marketing attack of the Atari-led Tramiels.

Commodore continues to insist that the Amiga is a business machine. One must assume that this is the reason none was present at CES. In the Commodore suite, only 64s and 128s were visible. It was simply amazing. And very quiet when we were there. The seeming lethargy in market positioning that has stricken Commodore since the introduction of the Amiga is one of the most shocking turnabouts we've witnessed in the modern history of this industry. One wonders whether the bankers have begun to call the strategic shots at Commodore.

We think that it is important to this industry as a whole that Commodore is, and continues to be, a viable player. Do not misunderstand. We saw nothing at CES that says it is not a viable company. We simply question the wisdom of its continued refusal to open up the Amiga market. Obviously such a decision is Commodore's, not ours, and obviously we're on the outside, but one can only marvel at the continued growth and success of Atari and the relative demise of the Amiga.

Last summer this time, both the ST and the Amiga were launched from an installed base of zero. Now, as we conclude the first year of product delivery, we find the ST with an installed base of roughly ten times that of the Amiga. Not a very stirring record. During this one-year period, the ST has grown, evolved, expanded to include the 1040, undergone in Tramiel-like fashion a predictable series of aggressive price cuts, expanded marketing outlets, etc. We've seen it all happen before with the VIC and the 64, but it's still quite impressive when it works.

Contrast with this the concurrent introduction of the Amiga. It was categorized, or defined, as a business machine. Its prices have changed only minimally. It has, to put it politely, withered. As we said, we think this industry needs Commodore, and it needs the vision and direction that a Commodore can help provide. We do not want it to be too late. Maybe if you gentlemen and ladies would just nudge the Amiga a little bit toward the consumer market, you'd be pleasantly surprised. Perhaps a price cut here, a market incursion there. You get the picture. You've got millions of users out here looking to you for technological leadership. Thanks. ©



Robert C. Lock
Editor in Chief

Flight Simulator II Scenery Disks

The Challenge of Accomplished Flight

With a realism comparable to (and in some ways even surpassing) $100,000 aircraft flight simulators, Flight Simulator II includes full flight instrumentation and avionics, and provides a full-color out-the-window view. Instruments are arranged in the format standard to modern aircraft. All the radios needed for IFR flight are included. Front, rear, left, right, and diagonal views let you look in any direction. Program features are clearly documented in a 96-page Pilot's Operating Handbook.

For training in proper flight techniques, Flight Simulator II includes another 96-page instruction manual, compiled by two professional flight instructors with over 8,000 hours flight time and 12,000 hours of aviation teaching experience. You'll learn correct FAA-recommended flight procedures, from basic aircraft control through instrument approaches. To reward your accomplishments, the manual even includes a section on aerobatic maneuvers.

The Realism and Beauty of Flight

Go sight-seeing over detailed, realistic United States scenery. High-speed graphic drivers provide an animated out-the-window view in either day, dusk, or night flying modes.

Flight Simulator II features over 80 airports in four different scenery areas: New York, Chicago, Seattle, and Los Angeles. Six additional Scenery Disks covering the entire Western half of the United States are now available in IBM and C64/128 disk formats.

Apple and Atari versions will be released soon. Each disk covers a geographical region of the country in detail, and is very reasonably priced.

The Pure Fun of "World War I Ace"

When you think you're ready, you can test your flying skills with the "World War I Ace" aerial battle game. This game sends you on a bombing run over heavily-defended enemy territory. Six enemy fighters will attempt to engage you in combat as soon as war is declared. Your aircraft can carry five bombs, and your machine guns are loaded with 100 rounds of ammunition.

See Your Dealer. Flight Simulator II is available on disk for the Apple II, Atari XL/XE, and Commodore 64/128 computers for $49.95. Scenery Disks for the C64 and IBM PC (Jet or Microsoft Flight Simulator) are $19.95 each. A complete Western U.S. Scenery six-disk set is also available for $99.95. For additional product or ordering information, call (800) 637-4983.

Apple II is a trademark of Apple Computer Inc
Atari XL and XE are trademarks of Atari Corp
Commodore 64 and 128 are trademarks of Commodore Electronics Ltd
IBM PC is a registered trademark of International Business Machines Corp

subLOGIC Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex 206995

Order Line: (800) 637-4983
(except in Illinois Alaska and Hawaii)

Publisher	James A. Casella
Founder/Editor In Chief	Robert C. Lock
Senior Editor	Richard Mansfield
Managing Editor	Kathleen Martinek
Executive Editor	Selby Bateman
Editor	Tom R. Halfhill
Assistant Editor	Philip Nelson
Production Director	Tony Roberts
Production Editor	Gail Cowper
Editor, COMPUTE!'s GAZETTE	Lance Elko
Technical Editor	Ottis R. Cowper
Assistant Technical Editor	George Miller
Program Editor	Charles Brannon
Assistant Editor, COMPUTE!'s GAZETTE	Todd Heimarck
Assistant Features Editor	Kathy Yakal
Programming Supervisor	Patrick Parrish
Editorial Programmers	Tim Victor, Kevin Mykytyn, Tim Midkiff
Submissions Reviewer	Mark Tuttle
Programming Assistants	David Florance, David Hensley
Executive Assistant	Debi Nash
Administrative Assistants	Julie Fleming, Ina Brooks, Mary Hunt, Sybil Agee
Associate Editors	Jim Butterfield, Toronto, Canada Harvey Herman, Greensboro, NC Fred D'Ignazio, Birmingham, AL David Thornburg, Los Altos, CA
Contributing Editor	Bill Wilkinson
COMPUTE!'s Book Division Editor	Stephen Levy
Assistant Editors	Gregg Keizer, Ann Davies
Director, Book Sales & Marketing	Steve Voyatzis
Production Manager	Irma Swain
Art & Design Director	Janice R. Fary
Assistant Editor, Art & Design	Lee Noel
Mechanical Art Supervisor	De Potter
Artists	Debbie Bray, Dabney Ketrow
Typesetting	Terry Cash, Carole Dunton
Illustrator	Harry Blair
Director of Advertising Sales	Peter Johnsmeyer
Associate Advertising Director	Bernard J. Theobald, Jr.
Production Coordinator	Kathleen Hanlon
Promotion Assistant	Caroline Dark
Customer Service Manager	Diane Longo
Dealer Sales Supervisor	Orchid Tamayo
Individual Order Supervisor	Cassandra Green
Receptionist	Anita Armfield
Warehouse Manager	John Williams

James A. Casella, President
Richard J. Marino, Vice President, Advertising Sales
Christopher M. Savine, Director, Finance & Planning

COMPUTE! Publications, Inc. publishes

COMPUTE!
The Journal For Progressive Computing

COMPUTE!'s GAZETTE
For VIC-20 and Commodore 64 Home Computers

COMPUTE! Books

COMPUTE!'s GAZETTE DISK

COMPUTE!'s Apple Applications Special

Editorial offices	324 West Wendover Avenue Suite 200 Greensboro, NC 27408 USA
Corporate offices	825 7th Avenue New York, NY 10019 212-265-8360
Customer Service	800-346-6767 (In NY 212-887-8525)
Hours:	9:30 A.M.–4:30 P.M. Monday–Friday

Coming In Future Issues

Beehive: An Exciting Strategy Game For Amiga, Commodore 64/128, Atari, Apple, And IBM PC/PCjr

Enhancements For Atari *SpeedCalc*

Dr. Sound For The 64/128

Apple PowerKey

Full-Screen Animation For IBM

A Full-Featured Home Financial Calculator For Atari ST

Amiga BASIC Style

Subscription Orders

COMPUTE!
P.O. Box 10954
Des Moines, IA 50340

TOLL FREE
Subscription Order Line
800-247-5470
In IA 800-532-1272

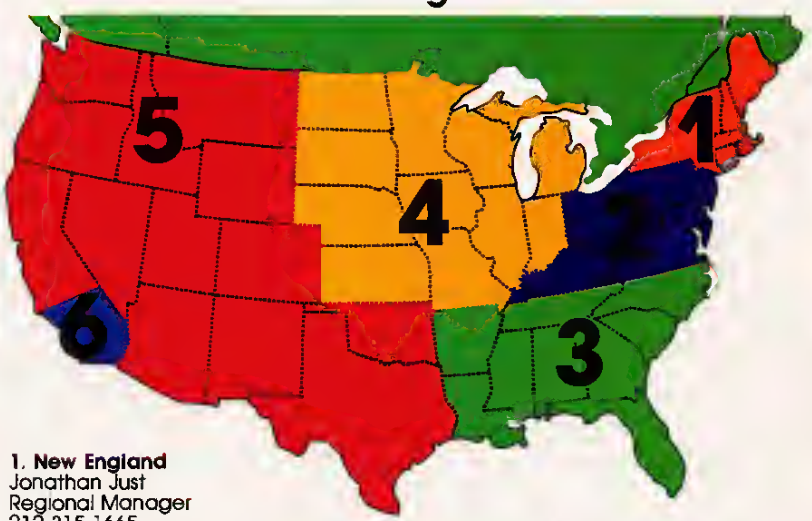
COMPUTE! Subscription Rates (12 Issue Year):

US	(one yr.)	$24
	(two yrs.)	$45
	(three yrs.)	$65
Canada and Foreign Surface Mail		$30
Foreign Air Delivery		$65

Member ABC Audit Bureau of Circulations

MPA Magazine Publishers Association

Advertising Sales



1. New England
Jonathan Just
Regional Manager
212-315-1665

2. Mid Atlantic
Jonathan Just
Regional Manager
212-315-1665

3. Southeast & Foreign
Harry Blair
919-275-9809

4. Midwest
Gordon Benson
312-362-1821

5. Northwest/ Mountain/Texas
Phoebe Thompson
Dani Nunes
408-354-5553

6. Southwest
Ed Winchell
213-378-8361

Director of Advertising Sales:
Peter Johnsmeyer

Associate Advertising Director:
Bernard J. Theobald, Jr.

COMPUTE! Home Office 212-887-8460.

Address all advertising materials to:
Kathleen Hanlon
Advertising Production Coordinator
COMPUTE! Magazine
324 West Wendover Avenue
Suite 200
Greensboro, NC 27408

The COMPUTE! subscriber list is made available to carefully screened organizations with a product or service which may be of interest to our readers. If you prefer not to receive such mailings, please send an exact copy of your subscription label to: COMPUTE! P.O. Box 10955, Des Moines, IA 50950. Include a note indicating your preference to receive only your subscription.

Authors of manuscripts warrant that all materials submitted to COMPUTE! are original materials with full ownership rights resident in said authors. By submitting articles to COMPUTE!, authors acknowledge that such materials, upon acceptance for publication, become the exclusive property of COMPUTE! Publications, Inc. No portion of this magazine may be reproduced in any form without written permission from the publisher. Entire contents copyright © 1986, COMPUTE! Publications, Inc. Rights to programs developed and submitted by authors are explained in our author contract. Unsolicited materials not accepted for publication in COMPUTE! will be returned if author provides a self-addressed, stamped envelope. Programs (on tape or disk) must accompany each submission. Printed listings are optional, but helpful. Articles should be furnished as typed copy (upper- and lowercase, please) with double spacing. Each page of your article should bear the title of the article, date and name of the author. COMPUTE! assumes no liability for errors in articles or advertisements. Opinions expressed by authors are not necessarily those of COMPUTE!

PET, CBM, VIC-20 and Commodore 64 are trademarks of Commodore Business Machines, Inc. and/or Commodore Electronics Limited.
Apple is a trademark of Apple Computer Company.
IBM PC and PCjr are trademarks of International Business Machines, Inc.
ATARI is a trademark of Atari, Inc.
TI-99/4A is a trademark of Texas Instruments, Inc.
Radio Shack Color Computer is a trademark of Tandy, Inc.

Another Great Simulation from Sid Meier – Author of F-15 Strike Eagle

Now he takes you from the cold, thin air and limitless space of F-15 Strike Eagle down into the dark depths of the Pacific Ocean inside an American World War II submarine for a realistic, action-filled simulation —

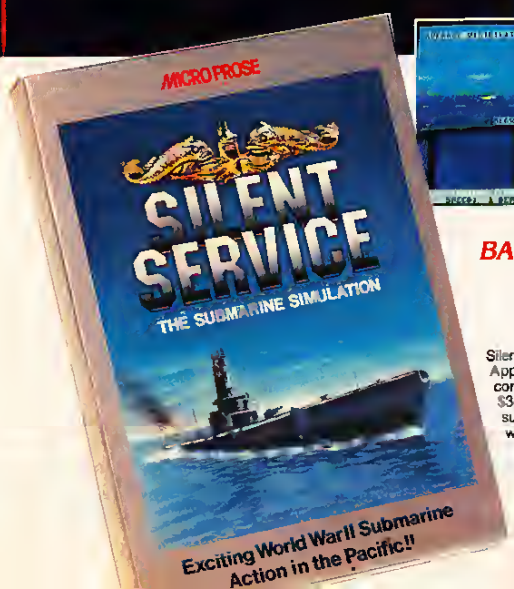
SILENT SERVICE

Thrill to the initial sighting of the enemy's strike force in your periscope as their ships come into your range. But watch out — the enemy's escorts have just sighted you. You're the **hunter** — but suddenly — you've become the **hunted!**

As Commander, you must sink their ships and keep your submarine from being destroyed — if you can. Will you select a quiet patrol sector in the Marianas Islands or choose the dangerous waters off the coast of Japan? Is a submerged daylight periscope attack best or do you charge in on the surface at night using only radar bearings to guide you? Do you fire a spread of your precious torpedoes or can you close the range and pick off the enemy with a single torpedo shot? These decisions and many more are yours to make as you take your place among the elite ranks of the **SILENT SERVICE!**

It's **exciting** — and it's **fun.** It's another great Micro Prose simulation — and it's called **SILENT SERVICE.** Look for it now on your dealer's shelves.

FIVE AUTHENTIC BATTLE STATION SCREENS



Silent Service is available for Commodore 64/128, Apple II Family, Atari XL/XE, IBM PC/PC Jr. computers for a suggested retail of only $34.95. Available soon for Macintosh for a suggested retail of only $39.95. Call or write for more information or MC/VISA orders

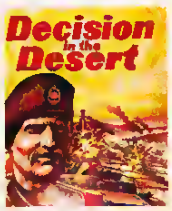
COMMODORE, APPLE, ATARI, IBM and MACINTOSH are registered trademarks of Commodore Business Machines Inc., Apple Computer Inc., Atari Inc., and International Business Machines Inc., respectively

Photo courtesy Baltimore Maritime Museum

Try These Other Real Life Simulations

Daring Attack Helicopter Action in the AH-64 Apache!

Thrilling Decathalon of aerobatic events in your own personal jet!



You are in command — North Africa 1940-1942

MICRO PROSE
SIMULATION • SOFTWARE
120 LAKEFRONT DRIVE • HUNT VALLEY, MD 21030 • (301) 667-1151

Readers Feedback

The Editors and Readers of COMPUTE!

If you have any questions, comments, or suggestions you would like to see addressed in this column, write to "Readers' Feedback," COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. Due to the volume of mail we receive, we regret that we cannot provide personal answers to technical questions.

Assemblers And Monitors

I am a little confused about the difference between an assembler and a machine language monitor. Would you please explain the functions of each?

Adam C. Stuart

Simply put, an assembler *is a program designed for one specific purpose—helping you write machine language programs. A* monitor *can be used for that purpose, too, but can also perform other memory-management tasks. Most programmers use an assembler for writing long ML programs and a monitor for writing short, experimental routines or debugging the code produced by an assembler.*

To illustrate the difference, let's say that you have a short machine language program beginning at location 491S2 ($C000) on the Commodore 64. A monitor allows you to examine the contents of any memory location. If you type M C000 C00C from the monitor, the following display might appear:

```
:C000  A9 42 20 06  C0 60 20  D2
:C008  FF 20 D2 FF  60 00 00  00
```

This memory display, like other monitor output, is in hexadecimal (base 16) notation. The numbers in the leftmost column are memory addresses; the numbers to the right show the actual contents of each successive location. Unless you're very familiar with hex notation and the 6502 instruction set, it's difficult to understand the program in this form. As a convenience, the monitor can translate machine language instructions from a series of raw numbers into more descriptive mnemonic *labels. This process is called* disassembly. *Here's how a monitor would disassemble the numbers seen in the display above:*

```
,C000  A9     42      LDA  #$42
,C002  20     06  C0  JSR  $C006
,C005  60             RTS
,C006  20     D2  FF  JSR  $FFD2
,C009  20     D2  FF  JSR  $FFD2
,C00C  RTS
```

Each three-letter mnemonic stands for a single ML instruction. In this example, the LDA (LoaD A) instruction loads the ASCII value for the letter B (hex $42) into the computer's A register, also called the accumulator. The JSR (Jump to SubRoutine) instruction calls a subroutine, much like GOSUB in BASIC. RTS (ReTurn from Subroutine) terminates a routine, much like RETURN in BASIC.

The converse of disassembly is assembly, *which lets you write a program by typing in mnemonics rather than numbers. To assemble the first line of the above program, for instance, you would type this line into the machine language monitor:*

```
A  C000  LDA  #$42
```

This puts the numbers $A9 and $42 into memory locations $C000 and $C001, where the computer interprets them as LoaD A with $42.

In addition to memory display, disassembly, and assembly, a monitor can perform other general tasks such as moving the contents of one memory area to another area, filling memory with a certain value, saving and loading an area of memory to tape or disk, and so forth. Monitors are so useful, in fact, that several computers, including the Commodore PET, Apple II, and Commodore 128, include one as a built-in feature.

An assembler, *as the name implies, is intended to do only one job—assemble an ML program from mnemonics. Since it usually can't disassemble the contents of memory, do memory moves, etc., an assembler is less versatile than a monitor. And programming with an assembler requires two steps instead of one. First you write a text file containing all the program instructions; this file is called the* source code. *Then you run the assembler, which translates the source code into executable* object code. *At first, the assembler sounds more cumbersome to use. But except for very short programs, it's considerably more convenient than a monitor. To illustrate, here is what the source code for this program might look like (this example is written in a format for the Commodore 64 PAL assembler; other assemblers are very similar):*

```
100 sys 700
110 .opt p4
120 *=49152
130 letter = 66
140 chrout = $ffd2
150 ; print 'b' twice
160 lda #letter
170 jsr print2
180 rts
190 print2 = *
200 jsr $ffd2:jsr $ffd2
210 rts
220 .end
```

*This assembler lets you write, save, and reload the ML source code as if it were a BASIC program, using sequentially numbered program lines. (Other assemblers provide similar functions.) Just as in BASIC, you can combine more than one statement on a single line (see line 200). Descriptive names can be given to constants (line 130), variables, ROM routines (140), and memory locations within the program itself (190). Assemblers also permit more flexibility of expression than monitors: Usually, decimal and hexadecimal numbers can be used interchangeably, and the assembler can evaluate strings and complex expressions as well. In this program, for instance, you can replace LDA #LETTER with LDA #$42, LDA #66, LDA #"B", or even LDA #(6+6*10).*

*The ability to use labels makes well-written assembler code much more readable than a monitor disassembly. The instruction JSR PRINT2, for instance, is more informative than JSR $C006. Labels also eliminate the need for tedious address calculations and simplify the process of relocating a program from one area of memory to another. When writing this program for an assembler, you don't need to know where the routine PRINT2 will actually end up in memory; the assembler handles such details for you automatically. With a monitor, on the other hand, you need to calculate the actual address of the subroutine before you can type in the JSR instruction that calls it. To move this program from location 49152 to 24576, you would simply change the origin statement in line 120 to *=24576 and reassemble the code. The assembler automatically adjusts everything to fit the new location.*

Even greater flexibility is offered through pseudo-ops *(pseudo-operations), which control various assembler functions. For instance, the .OPT pseudo-op*

USE THE BRAINS YOUR APPLE WASN'T BORN WITH.

Right at your fingertips in CompuServe's Apple® Forums.

Join the CompuServe **Apple II and III Forum** to swap everything from tall tales to short cuts with other users, and explore thousands of classic programs stockpiled since 1979.

Swap programs and files with fellow Mac owners in our **Macintosh® Users Forum.** Questions? You'll get answers from the experts here!

Visit the **Macintosh Developers Forum.** Get updates to the "Inside Macintosh Software Supplement." Interact with the Mac "team" in Cupertino.

The **Apple User Groups Forum,** supported by Apple Computer, unites officers of Apple user groups—"ambassadors" for hundreds of thousands of Apple-active enthusiasts worldwide.

Easy access to free software.

- *Download First-rate, non-commercial, user-supported software and utility programs.*
- *Take advantage of CompuServe's inexpensive weeknight and weekend rates (when Forums are most active, and standard online charges are just 10¢ a minute).*
- *Go online in most major metropolitan areas with a local phone call.*
- *And receive a* ***$25.00 Introductory Usage Credit*** *with purchase of your CompuServe Subscription Kit.*

Information you simply can't find anywhere else.

Use the Forum *Message Board* to send and receive electronic messages, and pose specific questions to Apple owners.

Join ongoing, real-time discussions in a Forum *Conference*—with Apple luminaries like Bill Atkinson, Doug Clapp, Dan Cochran, Jean-Louis Gassee, Mark Pelczarski, John Sculley and Steve Wozniak.

Search Forum *Data Libraries* for free software, user tips, transcripts of previous CompuServe online conferences and more.

Enjoy other useful services like:

- ***Popular Computer Magazines***—*electronic editions, for your reading pleasure. Including* ***Apples Online,*** *which reprints articles from leading user group newsletters nationwide and other Apple-related publications.*
- ***Other CompuServe Forums***—*supporting Jazz™ and other LOTUS® products. Microsoft®, MicroPro®, Borland International,® Ashton-Tate,® and other software. Also Pascal, Basic, C, Forth, Assembly and other programming languages.*

All you need is your Apple computer and a modem . . . or almost any other personal computer.

To buy your Subscription Kit, see your nearest computer dealer. Suggested retail price is $39.95. To receive our free brochure, or to order direct, call 800-848-8199 (in Ohio, call 614-457-0802). If you're already a CompuServe subscriber, type GO MAUG (Micronetworked Apple Users Group) at any ! prompt to see what you've been missing.

CompuServe®

Information Services, PO Box 20212
5000 Arlington Centre Blvd, Columbus, Ohio 43220

800-848-8199

In Ohio, Call 614-457-0802
An H&R Block Company

(line 110) tells the assembler where to send its output. By changing this instruction you can send output to memory, a disk file, the screen, or a printer. Assembling to a printer is particularly useful for making documentation, since the output includes everything you would see in a monitor disassembly (addresses, opcodes, and mnemonics) as well as all the comments and so forth contained in the assembler source file.

Other assembler pseudo-ops let you perform more advanced operations such as a conditional assembly, *which can include different segments of code (perhaps specific to various computers) in the assembly only when certain IF tests are satisfied. For long programs, a* linked assembly *allows you to assemble two or more separate source files into a single object file. The latter method was used to assemble the* SpeedScript *word processor, since the source code for that program is too long to fit in the computer's memory all at once.*

Atari BASIC Errors

Only recently have I become obsessed with home computers. As a novice, I decided to start with an Atari 400 (a 1982 model, I believe) and a cassette recorder. After many hours spent typing in your programs, I was constantly rewarded with error messages. I finally discovered that the BASIC cartridge accompanying the computer had since been revised twice. Not being able to locate the revision C cartridge in Dallas, I wrote Atari. No answer yet. Can you provide any insight? Also, is the 400 capable of using a disk drive, or am I stuck with tape?

Tom Rowan

It's true that Atari BASIC has been revised twice since your Atari was made, but it's unlikely that this is the source of your problems. The two revisions—known as revision B and revision C—mainly fix bugs in the original Atari BASIC cartridge. These bugs, however, don't cause spurious error messages. Usually they either lock up the computer entirely or mangle text strings. Your error messages are almost certainly due to mistakes in the programs, not problems with your BASIC.

You don't say whether the error messages appear when you're typing the programs or running the programs. Atari BASIC is one of the few BASIC languages that has instant syntax checking, so if you get an error immediately after typing a line and pressing RETURN, it usually means that a BASIC command was mistyped, a parenthesis was omitted, or the command is being used improperly. Examine the line carefully for any typos. If you can't find any, refer to the Atari BASIC Reference Manual *to see if the command usage is legal. For example, the statement A=CHR$(A$) generates an error because the CHR$ function is intended for converting a number into a string, and the variable A$ is already a string.*

Error messages encountered while you're trying to run a program are not due to syntax errors. Usually they indicate that the program is asking the computer to do something impossible. For example, the one-line program 10 GOTO 20 generates the message ERROR 12 AT LINE 10. If you look up this error number in the Atari BASIC Reference Manual, *it means Line Not Found. The command GOTO 20 tells the computer to branch to line 20, but this program has no line 20. (If you're typing in listings from* COMPUTE!, *you can avoid most of these mistakes by using our "Automatic Proofreader" program found in every issue.)*

If you'd still like a revision C Atari BASIC cartridge—worth having for only $15—you can order one from Atari. (Atari Corp., Customer Relations, 390 Caribbean Drive, Sunnyvale, CA 94088.) Be patient, though. It takes quite some time for Atari to fill these orders.

The Atari 400 is quite capable of using a disk drive if it has at least 16K of Random Access Memory (RAM). Early 400s had only 8K RAM. You can find out how much memory your 400 has by plugging in BASIC, typing NEW, and entering PRINT FRE(0). A 16K machine should return a number around 13000. However, we recommend at least 32K for use with a disk drive. A drive requires that you load a special program called a Disk Operating System (DOS), and this would consume more than half the available memory on a 16K system, leaving very little room for your BASIC program. The 400 can be upgraded to 48K or 64K, but the memory board installation isn't trivial. Also, for a few dollars more you could probably buy a new 800XL or 65XE.

Commodore 128 Sprites

I really enjoy programming with my new Commodore 128. However, using sprites has left me quite frustrated. The system guide's explanation of sprites doesn't explain how you can have more then eight sprite definitions in memory. Is there any way to do this?

Matt Lindquist

The Commodore 128 has room for only eight sprite shapes in its sprite definition area (memory locations 3584–4095). However, BASIC 7.0 includes a command (SPRSAV) which lets you move sprite shapes from strings into the sprite definition area and vice versa. Here is one form of SPRSAV:

SPRSAV 1,A$

This command moves the definition for sprite 1 into the string A$. Now the shape data is stored for later use. Here's the opposite form of SPRSAV:

SPRSAV A$,1

This command moves the shape data stored in A$ into the definition area for sprite 1. Of course, you can replace the name A$ with any legal string variable name.

The following program draws 16 sine waves on the screen, each positioned a little differently, then saves the sprites in the array A$ using the SSHAPE command. After all the shapes have been drawn and saved, sprite 1 is displayed on the screen. SPRSAV is then used to flip between the various sprite shapes. The rapid display of shapes makes the sine wave appear to move.

(Note: The underlined up-arrow (↑) in line 30 means to hold down the SHIFT key while pressing the up-arrow key. This will produce the pi (π) symbol.)

```
10 FAST
20 DIM A$(16)
30 FOR V=0 TO ↑*2 STEP ↑/8
40 GRAPHIC 1,1
50 FOR X=0 TO 23 STEP .2:Y=
   INT(11+10*SIN(X/2+V)):DR
   AW 1,X,Y:NEXT
60 SSHAPE A$(SN),0,0,23,20
70 SN=SN+1:NEXT V:GRAPHIC 0
   ,1:SLOW
80 SPRITE 1,1,2,1,1,1,0
90 MOVSPR 1,120,80:MOVSPR 1
   ,90#3
100 FOR A=0 TO 15:SPRSAV A$
    (A),1:FORT=1TO5:NEXT:NE
    XT A:GOTO 100
```

Apple Double Hi-Res Graphics

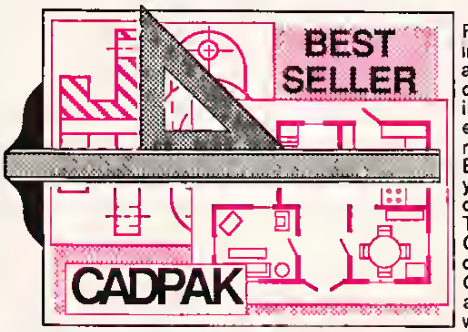
I'm having trouble understanding how the double high-resolution graphics mode works on my Apple IIc. How does the computer store the color and dot information? Is it possible to convert a normal hi-res picture to double hi-res format?

Robert Colello

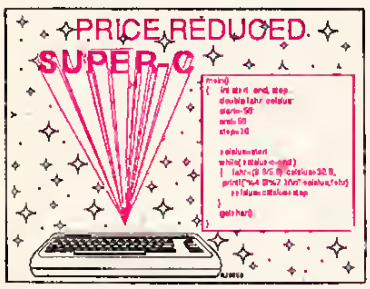
An Apple II that has 128K of Random Access Memory (any Apple IIc, or a IIe with extended 80-column card) can display pictures that have twice as many pixels across as normal hi-res pictures: that's 560 pixels in double hi-res versus the normal 280-pixel resolution. This display mode works in about the same way as 80-column text mode. For every byte of normal display memory, there's another byte with the same address in another bank *of memory, called* auxiliary RAM. *In normal hi-res mode, one byte of display data tells the computer how to draw seven pixels on the screen. In double hi-res, 14 dots can be drawn in the same space on the screen. The first seven dots are read*

'128™ and C-64™ PROVEN PERFORMANCE

The *complete* compiler and development package. Speed up your programs 5x to 35x. Many options: flexible memory management; choice of compiling to machine code, compact p-code or both '128 version. 40 or 80 column monitor output and FAST-mode operation. '128 Compiler's extensive 80-page programmer's guide covers compiler directives and options, two levels of optimization, memory usage, I/O handling, 80 column hi-res graphics, faster, higher precision math functions, speed and space saving tips, more. A great package that no software library should be without **128 Compiler $59.95**
64 Compiler $39.95

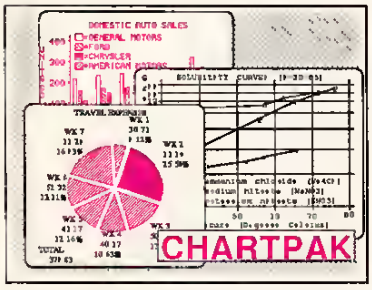


Remarkably easy-to-use interactive drawing package for accurate graphic designs. New dimensioning features to create exact scaled output to all major dot-matrix printers. Enhanced version allows you to input via keyboard or high quality lightpen. Two graphic screens for COPYing from one to the other. DRAW, LINE, BOX, CIRCLE, ARC, ELLIPSE available. FILL objects with preselected PATTERNS; add TEXT; SAVE and RECALL designs to/from disk. Define your own library of symbols/objects with the easy-to-use OBJECT MANAGEMENT SYSTEM–store up to 104 separate objects. **C-128 $59.95**
C-64 $39.95



For school or software development. Learn C on your Commodore with our in-depth tutorial. Compile C programs into *fast* machine language. C-128 version has added features: Unix™-like operating system; 60K RAM disk for fast editing and compiling Linker combines up to 10 modules; Combine M/L and C using CALL; 51K available for object code; Fast loading (8 sec. 1571, 18 sec. 1541); Two standard I/O librarys plus two additional libraries—math functions (sin, cos, sqrt, etc.) & 20+ graphic commands (line, fill, dot, etc.). **C-128 $59.95**
C-64 $59.95

Not just a compiler, but a complete system for developing applications in Pascal with graphics and sound features. Extensive editor with search, replace, auto, renumber, etc. Standard J & W compiler that generates fast machine code. If you want to learn Pascal or to develop software using the best tools available–**SUPER Pascal** is your first choice. **C-128 $59.95**
C-64 $59.95



Easily create professional high quality charts and graphs without programming. You can immediately change the scaling, labeling, axis, bar filling, etc. to suit your needs. Accepts data from CalcResult and MultiPlan. C-128 version has 3X the resolution of the '64 version. Outputs to most printers. **C-128 $39.95**
C-64 $39.95

OTHER TITLES AVAILABLE:

PowerPlan

One of the most powerful spreadsheets with integrated graphics. Includes menu or keyword selections, online help screens, field protection, windowing, trig functions and more. PowerGraph, the graphics package, is included to create integrated graphs and charts. **C-64 $39.95**

COBOL Compiler

Now you can learn COBOL, the most widely used commercial programing language, and learn COBOL on your 64. COBOL is easy to learn because its easy to read. COBOL Compiler package comes complete with Editor, Compiler, Interpreter and Symbolic Debugger. **C-64 $39.95**

BEST SELLER

Personal Portfolio Manager

Complete portfolio management system for the individual or professional investor. Easily manage your portfolios, obtain up-to-the-minute quotes and news, and perform selected analysis. Enter quotes manually or automatically through Warner Computer Systems. **C-64 $39.95**

Xper

XPER is the first "*expert system*" for the C-128 and C-64. While ordinary data base systems are good for reproducing facts, XPER can derive knowledge from a mountain of facts and help you make expert decisions. Large capacity. Complete with editing and reporting. **C-64 $59.95**

Technical Analysis System for the C-64	$59.95
Ada Compiler for the C-64	$39.95
VideoBasic Language for the C-64	$39.95

C-128 and C-64 are trademarks of Commodore Business Machines Inc
Unix is a trademark of Bell Laboratories

Abacus Software

P.O. Box 7219 Dept. C8 Grand Rapids, MI 49510 - Telex 709-101 - Phone (616) 241-5510

Call **now** for the name of your nearest dealer. Or to order directly by credit card, MC, AMEX of VISA call **(616) 241-5510**. Other software and books are available–Call and ask for your **free** catalog. Add $4.00 for shipping per order. Foreign orders add $12.00 per item. Dealer inquires welcome–1400+ nationwide.

from auxiliary memory and the second set comes from main RAM.

One double hi-res screen occupies 16K of memory between addresses 8192–16383 ($2000–$3FFF) in each bank. Unlike standard hi-res, there's only one double hi-res screen, so it's not practical to create animation with page flipping.

Here are some routines that will help you get started with double hi-res graphics. When run, they create machine language programs called DCONVERT *and* DHGRSAVE. *If you load a normal hi-res picture into hi-res screen 1 (at 8192) then BRUN the* DCONVERT *program, it converts the picture to double hi-res format and displays it.*

To save this or any other double hi-res picture to disk, BRUN the DHGRSAVE program, then enter BSAVE filename *,A$2000,L$4000 (replace* filename *with the name of your choice). The graphics image is saved in the same format used by* Dazzle Draw *and other double hi-res programs. If you save the image file on a ProDOS disk, you can then load it with* Dazzle Draw *and modify the picture.*

```
80 FOR I = 24576 TO I + 161:
   READ A: POKE I,A: NEXT
90 PRINT CHR$ (4);"BSAVE DCON
   VERT,A$6000,L$A2"
100 DATA 141,126,192,173,94,1
   92,173,87,192,173,82
110 DATA 192,141,13,192,141,0
   ,192,173,80,192,169
120 DATA 0,133,254,169,32,133
   ,255,160,0,177,254
130 DATA 72,72,41,15,170,189,
   146,96,141,163,96
140 DATA 104,74,74,74,74,41,7
   ,170,189,146,96
150 DATA 141,162,96,104,16,11
   ,173,163,96,201,128
160 DATA 173,162,96,42,144,27
   ,78,163,96,136,48
170 DATA 10,177,254,8,10,10,4
   0,106,74,145,254
180 DATA 200,173,162,96,44,59
   ,96,240,2,9,64
190 DATA 145,254,173,163,96,4
   1,127,141,5,192,145
200 DATA 254,141,4,192,200,19
   2,40,208,168,165,254
210 DATA 105,127,133,254,144,
   2,230,255,165,255,201
220 DATA 64,208,150,165,254,1
   05,39,133,254,201,120
230 DATA 208,136,96,0,3,12,15
   ,48,51,60,63
240 DATA 192,195,204,207,240,
   243,252,255
```

```
80 FOR A = 4096 TO A + 47: RE
   AD I: POKE A,I: NEXT
90 PRINT CHR$ (4);"BSAVE DHGR
   SAVE,A$1000,L$30"
100 DATA 160,0,132,252,132,25
   4,169,32,133,253,169
110 DATA 64,133,255,141,1,192
   ,173,87,192,177,252
120 DATA 145,254,141,85,192,1
   77,252,141,84,192,145
130 DATA 252,200,208,239,230,
   253,230,255,165,253,201
140 DATA 64,208,229,96
```

Saving PCjr Screens

I have been experiencing trouble with BSAVEing SCREEN 5 on my PCjr. For some reason, the computer loads only half the picture when I try to bring it back into memory.

Marc Ramirez

The PCjr was designed to be as compatible as possible with the IBM PC, but there are several differences, most notably the lack of DMA *(Direct Memory Access) hardware that speeds certain operations on the PC. On the other hand, the PCjr has better color graphics than the PC. Its SCREEN 5 mode gives you 320 × 200 resolution with 16 simultaneous screen colors. These don't represent fixed colors as in the PC-compatible modes. Instead, each of the 16 colors can be redefined to use any of the 16 possible colors, making available the advantages of color indirection.*

The IBM PC color/graphics card contains 16K of onboard RAM for its own use. Because the RAM is part of the color card, there is no conflict when both the screen and the microprocessor want to access memory at the same time. However, references to addresses $B8000–$BC000 are redirected to the color card's memory, which permits the microprocessor to update screen memory and redraw the screen directly.

The PCjr, however, has no memory at $B8000. Screen memory is taken from the main store of RAM, usually at location $18000. This explains why the PCjr is slower than its big brother. The graphics chips need to access screen memory constantly while building the screen, and since this memory is on the main address bus, the microprocessor can't get at memory to execute instructions while the graphics chips are using it.

However, IBM realized that many commercial programs try to update the screen by storing values directly into screen memory at $B8000. To keep the PCjr compatible with these programs, IBM modified the address circuitry to redirect references to $B8000 to the actual screen memory area. However, only 16K of memory is redirected. Since a SCREEN 5 screen is 32K long, this explains why you're seeing only half of the picture.

When you use the sequence

```
DEF SEG=&HB800:BSAVE "screen" ,0
   ,32768!
```

the first 16K of memory is saved from the area at $18000, but the rest of the picture is saved from $BD000–$C0FFF, since this memory range is not relocated. This second half is just whatever random bits are read when this nonexistent memory is saved. Instead, you need to use

```
DEF SEG=&H1800:BSAVE "screen",0
   ,32768!
```

to save the screen, and

```
DEF SEG=&H1800:BLOAD "screen"
```

to load it back. If you try to load images saved from the original range of $B8000–$C0FFF, the second interleaved half of the picture will be garbage. If you use two or more graphics screens, the additional screens are stored behind the first one at lower memory locations. The first SCREEN 5 screen would be at $18000, the second would be stored at $10000, and so forth.

TurboDisk With *64 SpeedScript*

Now that the commented source code for *SpeedScript* is available in book form, I have found ways to make the program work in all kinds of situations. Here are a couple of *SpeedScript* modifications I have found very useful.

Only two POKEs are needed to allow you to use "TurboDisk" (the fast-load utility published in the April 1985 issue of COMPUTE!) with *SpeedScript*. First, load in your copy of *SpeedScript* (version 3.0 or higher). Now enter these POKEs in direct mode (without line numbers). Be sure to press RETURN after typing each line:

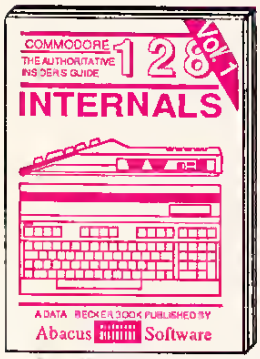
```
POKE 2481,191
POKE 4938,8
```

Now resave the program, using a different filename (perhaps SPEEDSCRIPT.TURB) to differentiate it from the original. To use the modified program, simply activate TurboDisk as usual, then load and run *SpeedScript*. You'll find that text files are loaded much faster than usual. If you exit *SpeedScript*, you must reactivate TurboDisk with SYS 49152.

A second useful change has to do with word wrap—*SpeedScript's* ability to automatically move a word down to the next line when it's too big to fit on the current line. Word wrap is great for making text readable, but creates headaches when you need to align the right margin or line up decimal points past the fortieth column. The following program replaces *SpeedScript's* Verify command (which I have never used) with a function that toggles word wrap on and off. Type in the following program and save a copy, then run it and follow the prompts (tape users note the line change below).

```
10 FORQ=49152TO49198:READA:
   X=X+A:NEXT:IFX<>6412THEN
    PRINT"ERROR IN DATA":EN
   D
20 RESTORE:FORQ=49152TO4919
   8:READA:POKEQ,A:NEXT
30 PRINT"{CLR}{WHT}LOAD YOU
   R VERSION OF"
40 PRINT"SPEEDSCRIPT 3.0 OR
    HIGHER"
50 PRINT"THEN SYS49152 AND
```

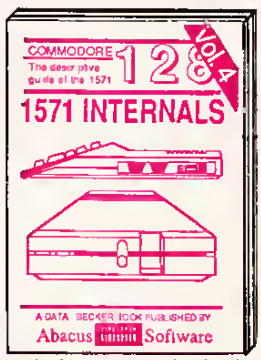
C-128™ and C-64™ AUTHORITATIVE BOOKS



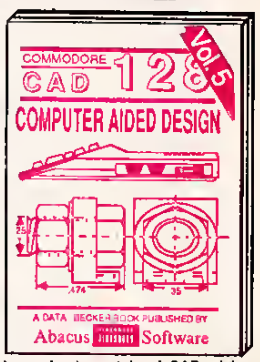
Detailed guide presents the 128's operating system, explains graphic chips, Memory Management Unit, 80 column graphics and commented ROM listings **500pp** $19.95

Get all the inside information on BASIC 7.0. This exhaustive handbook is complete with commented BASIC 7.0 ROM listings. Coming Summer '86 $19.95

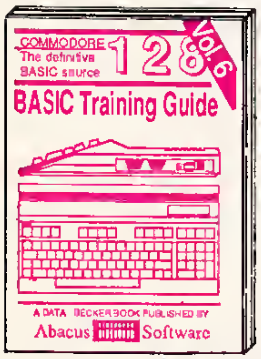
Filled with info for everyone. Covers 80 column hi-res graphics, windowing, memory layout, Kernal routines, sprites, software protection, autostarting **300pp** $19.95



Insiders' guide for novice & advanced users. Covers sequential & relative files, & direct access commands. Describes DOS routines. Commented listings. $19.95

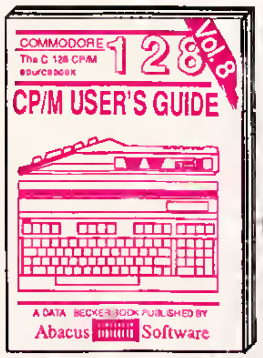


Learn fundamentals of CAD while developing your own system. Design objects on your screen to dump to a printer. Includes listings for '64 with Simon's Basic. **300pp** $19.95

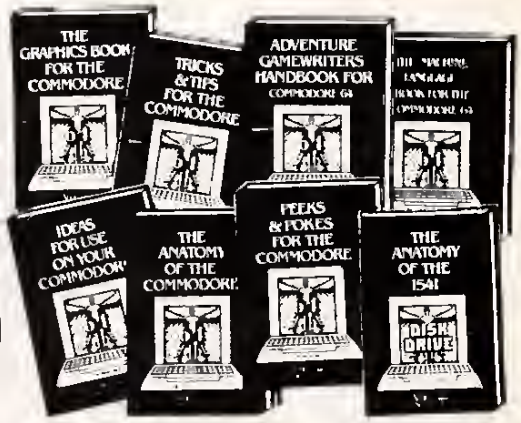


Introduction to programing, problem analysis, thorough description of all BASIC commands with hundreds of examples, monitor commands, utilities, much more $16.95

Presents dozens of programming quick-hitters. Easy and useful techniques on the operating system, stacks, zero page, pointers, the BASIC interpreter and more $16.95



Essential guide for everyone interested in CP/M on the 128. Simple explanation of the operating system, memory usage, CP/M utility programs, submit files & more $19.95



ANATOMY OF C-64 Insider's guide to the '64 internals. Graphics, sound, I/O, kernal, memory maps, more. Complete commented ROM listings. **300pp** $19.95

ANATOMY OF 1541 DRIVE Best handbook on floppy, explains all. Many examples and utilities. Fully commented 1541 ROM listings. **500pp** $19.95

MACHINE LANGUAGE C-64 Learn 6510 code write fast programs. Many samples and listings for complete assembler, monitor, & simulator. **200pp** $14.95

GRAPHICS BOOK C-64 - best reference covers basic and advanced graphics. Sprites, animation, Hires, Multicolor, lightpen, 3D-graphics, IRQ, CAD, projections, curves, more. **350pp** $19.95

TRICKS & TIPS FOR C-64 Collection of easy-to-use techniques: advanced graphics, improved data input, enhanced BASIC, CP/M, more. **275pp** $19.95

1541 REPAIR & MAINTENANCE Handbook describes the disk drive hardware. Includes schematics and techniques to keep 1541 running. **200pp** $19.95

ADVANCED MACHINE LANGUAGE Not covered elsewhere: - video controller, interrupts, timers, clocks, I/O, real time, extended BASIC, more. **210pp** $14.95

PRINTER BOOK C-64/VIC-20 Understand Commodore, Epson-compatible printers and 1520 plotter. Packed: utilities; graphics dump; 3D-plot; commented MPS801 ROM listings, more. **330pp** $19.95

SCIENCE/ENGINEERING ON C-64 In depth intro to computers in science. Topics: chemistry, physics, biology, astronomy, electronics, others. **350pp** $19.95

CASSETTE BOOK C-64/VIC-20 Comprehensive guide; many sample programs. High speed operating system fast file loading and saving. **225pp** $14.95

IDEAS FOR USE ON C-64 Themes: auto expenses, calculator, recipe file, stock lists, diet planner, window advertising, others. Includes listings. **200pp** $12.95

COMPILER BOOK C-64/C-128 All you need to know about compilers: how they work; designing and writing your own; generating machine code. With working example compiler. **300pp** $19.95

Adventure Gamewriter's Handbook Step-by-step guide to designing and writing your own adventure games. With automated adventure game generator. **200pp** $14.95

PEEKS & POKES FOR THE C-64 Includes in-depth explanations of PEEK, POKE, USR, and other BASIC commands. Learn the "inside" tricks to get the most out of your '64. **200pp** $14.95

Optional Diskettes for books
For your convenience, the programs contained in each of our books are available on diskette to save you time entering them from your keyboard. Specify name of book when ordering. **$14.95 each**

C-128 and C-64 are trademarks of Commodore Business Machines Inc.

Abacus Software

P.O. Box 7219 Dept. C8 Grand Rapids, MI 49510 - Telex 709-101 - Phone (616) 241-5510

Optional diskettes available for all book titles - $14.95 each. Other books & software also available. Call for the name of your nearest dealer. Or order directly from ABACUS using your MC, Visa or Amex card. Add $4.00 per order for shipping. Foreign orders add $10.00 per book. Call now or write for your free catalog. Dealer inquires welcome--over 1400 dealers nationwide.

```
   {SPACE}RUN."
49152 DATA 162,35,189,12,19
      2
49157 DATA 157,26,20,202,16
49162 DATA 247,96,173,219,8
49167 DATA 201,177,208,17,1
      69
49172 DATA 169,141,219,8,16
      9
49177 DATA 32,141,220,8,165
49182 DATA 197,201,31,240,2
      50
49187 DATA 96,169,177,141,2
      19
49192 DATA 8,169,251,141,22
      0
49197 DATA 8,96
```

After POKEing a short ML routine into memory, the program instructs you to load *SpeedScript* (3.0 or higher), enter SYS 49152, then run *SpeedScript*. Try toggling word wrap on and off by pressing CTRL-V (ordinarily the Verify function).

If you use tape instead of disk, you may not want to give up the Verify function but can easily afford to live without the Directory command, which is useless with tape anyway. In line 49157 of the program, change the 26 to 98. Then change the checksum value in line 10 from 6412 to 6484.

When you're satisfied that the modification works, exit *SpeedScript*. Disk users should enter POKE 2895,23 to change the Verify command from CTRL-V to CTRL-W (for Word wrap). Tape users should enter POKE 2898,23 to change the Directory command from CTRL-4 to CTRL-W. After that's done, resave *SpeedScript* under a new filename that reflects the change.

Bruce S. Gordon

Thanks for the suggestions. Incidentally, the penalty you pay for turboloading with SpeedScript *is that available text memory is reduced from 43,445 characters to 39,299 characters.*

Improved Atari Line Delete

Like many BASIC programmers, I usually number my programs with an increment of 10. Often, however, after editing and debugging, there is no longer any pattern to line numbering. This short utility program has a little more versatility than "Line Deleter For Atari," published in the January 1986 issue of COMPUTE!. As in the former, LIST the utility to disk or cassette, then load your BASIC program and ENTER this utility. Type GOTO 32700 in direct mode, then input the beginning and end range to be deleted. You can now delete only existing line numbers. When the deletion is finished, press RETURN to remove the utility from your BASIC program.

```
32700 REM BLOCK DELETE EN
      TER AND GOTO 32700
32701 TRAP 32713:? "START
      ,END":INPUT START,E
      ND
32702 ? CHR$(125):X=PEEK(
      138)+PEEK(139)*256
32703 B=PEEK(136)+PEEK(13
      7)*256:X=B:QQ=0:POS
      ITION 2,2
32704 LN=PEEK(X)+PEEK(X+1
      )*256
32705 IF LN<START THEN X=
      X+PEEK(X+2):GOTO 32
      704
32706 IF LN=32700 THEN 32
      710
32707 ? LN:QQ=QQ+1:IF QQ=
      18 THEN 32710
32708 IF LN>=EN THEN 3271
      0
32709 X=X+PEEK(X+2):GOTO
      32704
32710 TRAP 32713:? "32700
       REM PRESS RETURN T
      O REMOVE BLOCK DELE
      TER":? "CONT"
32711 POKE 842,13:POSITIO
      N 2,0:STOP
32712 POKE 842,12:GOTO 32
      700
32713 ? CHR$(125):POSITIO
      N 2,2:FOR SS=32700
      TO 32714:? SS:NEXT
      SS:? "POKE 842,12"
32714 POKE 842,13:POSITIO
      N 2,0:STOP
```

Gary Rindosh

Thank you for the program.

Dvorak Keyboard For 64

After 25 years of typing the "qwerty" way, I'd like to take advantage of a Dvorak keyboard toggle included in a *SpeedScript* enhancement program for the Commodore 64. What resources are available to help me learn the Dvorak system? Are keyboard caps for the 64 available so that I can cover up the normal keys with Dvorak caps? It's going to be hard giving up the old system, but everything I've heard about the speed and efficiency of the Dvorak keyboard makes me eager to give it a try.

John Willis

If your enhancement program can emulate the Dvorak keyboard within SpeedScript, *then no hardware is required to convert from the conventional typewriter key arrangement—often called qwerty—to the Dvorak scheme. Many office supply stores carry stick-on keycap labels that should suit your needs. We're assuming that you have a diagram which shows the Dvorak keyboard.*

The advantage of stick-on labels is that you can still use the computer for other purposes that don't involve a Dvorak keyboard. Most commercial software and virtually all type-in programs in publications like COMPUTE! *assume that you have a normal 64 keyboard. If you can find or fabricate blank stick-on labels, you could divide each label into two segments—indicate the Dvorak key on one half and the normal 64 key on the other. This would allow you to switch from Dvorak to qwerty applications at will.*

If you don't use the computer for anything other than word processing and decide to convert permanently to the Dvorak scheme, you could rearrange the existing keycaps. This operation doesn't require any special tools or electronics expertise. While you have the old keycaps off, you can take advantage of the opportunity to clean the keyboard, too. However, rearranging the keycaps will void any warranty that may be in effect, since you must open the case of the computer. And though the operation is reversible, you should consider it semipermanent because of the time involved in switching the keycaps.

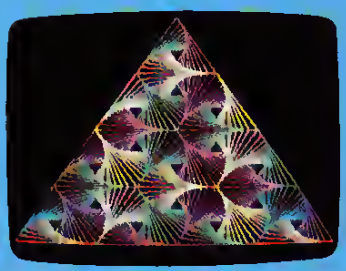
To rearrange the keycaps, remove the three Phillips screws in the bottom of the computer's case, then gently separate the two halves of the case. Carefully unplug the two sets of wires that join the upper and lower halves, then remove the eight Phillips screws that hold the keyboard assembly to the upper half. The 64's keycaps are held on by friction, so you can lever them off using a thin-bladed screwdriver or similar device. The alphanumeric keycaps are all the same size and can be interchanged freely. Of course, you shouldn't disturb keys such as RESTORE, which serve a special purpose. While the keycaps are off, you may want to clean the area around each keyswitch. In many cases, cleaning is all that's needed to fix keys that stick or bounce (repeat when they shouldn't).

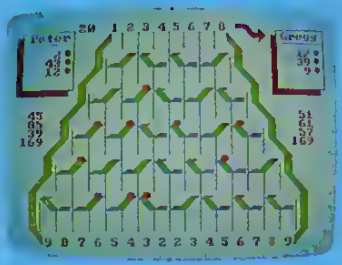
To replace a keycap, press it gently but firmly onto the shaft of the keyswitch. After all the keycaps are back in place, reverse the disassembly procedure: Screw the keyboard assembly back into the upper half of the case, then replace the two sets of wires that join the halves. Finally, rejoin the two halves of the case, turn the computer over, and replace the three screws on the bottom. If you've never performed the operation before, you should plan to spend a couple of hours removing, cleaning, and replacing the keycaps.

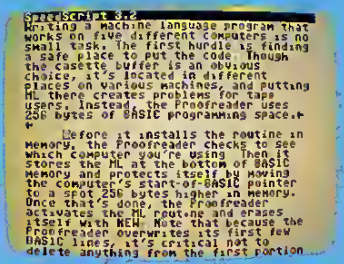
By the way, you might be interested to learn that there is some controversy surrounding the efficiency claims for the Dvorak keyboard. Most of the frequently quoted statistics (like 35–50 percent increase in speed and 90 percent reduction in finger travel) come from August Dvorak's own research. An independent investigation by Donald Olson and Laurie Jasinski, published in the February 1986 issue of BYTE *magazine, suggests that these figures are inflated. While agreeing that the Dvorak arrangement is somewhat*

GET UP TO 200 FUN-FILLED PROGRAMS EACH YEAR—

when you subscribe now to COMPUTE!







Subscribe to COMPUTE! today through this special introductory money-saving offer, and you'll be getting a lot more than just another computer magazine. That's because each issue of COMPUTE! comes complete with up to 20 all-new, action-packed programs.

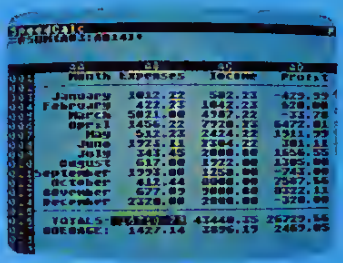
Subscribe now and you can depend on a steady supply of high quality, fun-filled programs like Hickory Dickory Dock, Switchbox, TurboDisk, Home Financial Calculator, Turbo Tape, SpeedScript, SpeedCalc, and hundreds of other educational, home finance, and game programs the entire family can use all year long.

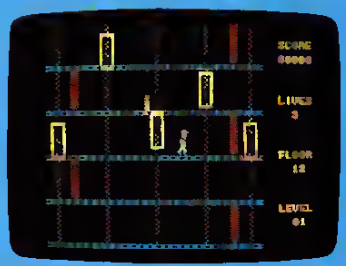
The superb programs you'll find in each issue are worth much, much more than the low subscription price.

And there's more to COMPUTE! than just exciting new programs. Month after month, COMPUTE!'s superb articles deliver the latest inside word on everything from languages to interfaces...from programming to disk drives.

Whether you're a novice or an experienced user, COMPUTE! is the magazine for you. So subscribe today. Return the enclosed card or call 1-800-247-5470 (in Iowa, 1-800-532-1272).

Do it now.





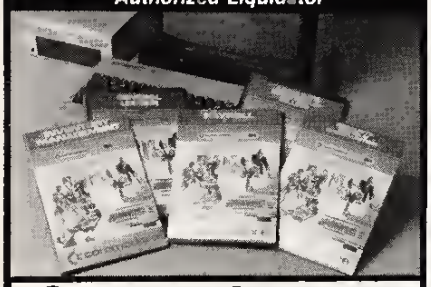
ACT NOW AND SAVE!

COMPUTE!™ Publications, Inc. abc

One of the ABC Publishing Companies

If attached order card is missing, write: COMPUTE!'s Gazette P.O. Box 10955, Des Moines, IA 50950

COMB
Authorized Liquidator



Commodore 64 BUSINESS SOFTWARE

A 5-pack of most needed software for efficient business operations!

Genaral Ledger
- Has 8 general ledger options.
- Provides 150 chart-of-accounts.
- 1500 general journel transactions

Inventory Management
- Tracking of 1000 inventory items.
- Maintains perpetual inventory records.
- Calculates use, reorders, cost averaging, etc

Payroll*
- Provides 24 different payroll functions.
- Calculates payroll and tax deductions.
- Ideal for 50 employees or less.

Accounts Receivable/Billing*
- Provides customer sales, credit information, printed statements and more.
- Handles 11 billing functions, 150 invoices, 75 customers

Accounts Payable/Checkwriting*
- Combines tracking of vendor payables with an integrated checkwriting system
- Maintains master file, provides invoice listings

*Interfaces with General Ledger software

90-Day Limited Factory Warranty

Mfr. List: $249.75
ENTIRE SET of 5
Liquidation Price $49
Item H-1377-7002-066 Ship, handling: $5.00

NOTE: Also available by individual titles. Phone for prices.

Credit card customers can order by phone, 24 hours a day, 7 days a week. VISA MasterCard AMERICAN EXPRESS

Toll-Free: 1-800-328-0609

Sales outside the 48 contiguous states are subject to special conditions. Please call or write to inquire.

C.O.M.B. Direct Marketing Corp. Item H-1377
14605 28th Ave. N./Minneapolis, MN 55441-3397

Send ___ 5-Pack(s) of Commodore 64 Business Software Item H-1377-7002-066 at $49 each plus $5 each for ship, handling (Minnesota residents add 6% sales tax. Sorry, no C.O.D. orders)

☐ My check or money order is enclosed (No delays in processing orders paid by check, thanks to TeleCheck)

Charge ☐ VISA® ☐ MasterCard® ☐ American Express®

Acct. No. _____ Exp. ___/___

PLEASE PRINT CLEARLY

Name _____

Address _____

City _____

State _____ ZIP _____

Phone ( ) _____

Sign Here _____

COMB Direct Marketing Corp.
Authorized Liquidator
14605 28th Avenue North
Minneapolis, Minnesota 55441-3397

more efficient, that article reports that the actual reduction in finger travel is less than 30 percent. It also quotes a University of California study which concluded that a speed increase of 5–10 percent was more realistic.

Automatic IBM Screen Printing

Some time ago I created a BASICA program which allows me to create graphics pictures much like a graphics editor. The program uses every graphics function in the manual, and even saves your work. But in order to print my creations on the printer, I have to press Shift-PrtSc. Is there any way to add a Print Screen function to my program?

William Green

Fortunately, it's quite easy to call the ROM BIOS routine that supports screen printing. The following program fragment does the trick by POKEing a tiny machine language program into a reserved space at the top of BASIC's memory area. The ML just executes INT5:RETF to call the Print Screen routine and return to BA5IC. This program is adapted from COMPUTE!'s Mapping the IBM PC and PCjr, *by Russ Davies.*

When incorporating this routine into your program, the line with the CLEAR statement must be the first line in your program. Otherwise, any previously defined variables will be erased. Once the machine language is POKEd into memory, your program can execute the statement CALL PRT5C to make a printout.

```
100 CLEAR ,&HFFF0:PRTSC=&HFFF
    0
110 DEF SEG:FOR X=0 TO 2:READ
    N:POKE X+PRTSC,N:NEXT
120 DATA &HCD,&H05,&HCB
200 CALL PRTSC
```

Atari DOS 3.0 Vs. 2.5

I have purchased an Atari 1050 disk drive with DOS 3.0. I recently heard that this DOS is no good, and that I should use DOS 2.0 or 2.5. What is so wrong with DOS 3.0, and why shouldn't I use it? Is DOS 2.5 the best one yet for the 1050, and where can I get it?

Gary Cerasoli

Before getting to your questions, let's briefly review the history of Atari disk operating systems:

• DOS 1.0 was introduced with the 400/800 computers and 810 disk drive in 1979. It was workable, but suffered from some bugs and unimplemented features. Also, the entire DOS was always resident in RAM (Random Access Memory). Although this was convenient—the DOS menu appeared instantly when you typed the DOS command—it consumed too much memory in a period when few people had more than 24K or 32K of RAM.

• DOS 2.0, also known as 2.0S (single-density), replaced DOS 1.0 in late 1980/early 1981. It fixed the bugs in DOS 1.0, added significant new features, and conserved memory by keeping only part of itself in RAM. The disk-resident portion of DOS 2.0 loads into memory only when you type the DOS command.

• DOS 3.0 was introduced with the dual-mode 1050 disk drive in 1983. The 1050 works in the traditional Atari single-density mode (88K of storage per disk) as well as an enhanced-density mode (127K of storage per disk). DOS 3.0 was designed to support the enhanced-density mode and to be easier to use. But most Atari users found DOS 3.0 to be clumsy and inconvenient, especially when swapping disks with other people or when mixing single-density and enhanced-density disks. Although the 1050 drive automatically adjusts itself for either density, DOS 3.0 disks and 2.0 disks are incompatible with each other.

• To solve these problems, DOS 2.5 was introduced in 1985. This numbering scheme sometimes confuses people, since 2.5 was released two years after 3.0, but 2.5 is so named because it is closely related to DOS 2.0. In fact, the 2.5 menu is almost identical to the 2.0 menu, save for one additional option (Format Single). The advantage of 2.5 is that it works with both single- and enhanced-density disks on the 1050 drive as well as single-density disks on the older 810 drives. This makes life easier for people who have both formats in their disk libraries and for those who swap disks with other users.

DOS 2.5 is available free from most Atari dealers and user groups. It comes with utilities for converting 3.0 files to 2.0/2.5 format, for customizing your copy of 2.5, and for automatically booting up a RAM disk on the 130XE computer.

There's a chance that 2.5 may be superseded in the near future by yet another DOS. Atari is thinking about introducing a 3½-inch disk drive for the 400/800/XL/XE line, and the much greater capacity of this format (at least 320K per side) would require a completely new DOS with support for subdirectories and other advanced file-management features. ©

"If you know BASIC and want to learn machine language, this is the place to start . . . . Building on your experience as a BASIC programmer, Mansfield very gently takes you through the fundamentals of machine language."

—Whole Earth Software Catalog

COMPUTE! Books' Best-selling Machine Language Books

"Understandable"—**The New York Times**

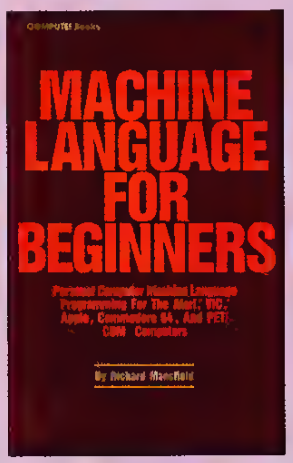
"Presente the machine language novice with a very good tutorial in simple, understandable terms."

—**Antic**

"I highly recommend *Machine Language for Beginners* ae your firet introduction to the world of machine language." —**Commodore Power/Play**

The LADS Disk

LADS, the ossembler used in *The Second Book of Mochine Longuoge*, is ovoilable on disk for only $12.95. This is a great accompaniment to the book, saving you hours of typing time by providing the complete source ond object programs for oll versions of the ossembler, and more. And LADS disks are specific to your Apple, Atari, or Commodore computers.



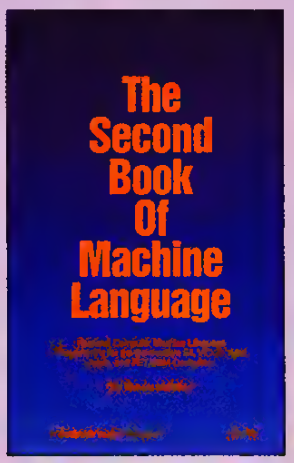
Machine Language for Beginners

Richard Mansfield

Most commercial software is written in machine language because it's far faster and more versatile than BASIC. *Machine Language for Beginners* is a step-by-step introduction. Includes a simple assembler, a disassembler, and utilities, to help beginners write programs more quickly and easily

$14.95

ISBN 0-942386-11-6



The Second Book of Machine Language

Richard Mansfield

The follow-up to the best-selling *Machine Language for Beginners*, this book leads the programmer deeper into the most powerful and efficient programming techniques available for personal computers. Fully tutorial, with easy step-by-step explanations, the book shows how to construct significant, effective machine language programs. Included is a high-speed, professional-quality, label-based assembler. Everything that's needed for optimized programming on the Commodore 64, Apple, Atari, VIC-20, and PET/CBM computers.

$14.95

ISBN 0-942386-53-1

Machine Language for Beginners and *The Second Book of Machine Language*: everything you need to learn machine language programming on your Apple, Atari, and Commodore personal computers.

To Order: Call Toll Free 800-346-6767 (In NY 212-887-8525) or mail this coupon with your payment to COMPUTE! Books, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

_____ *The Second Book of Machine Language*, **$14.95**
_____ *Machine Language for Beginners*, **$14.95**
_____ *LADS Disk* (Apple) **$12.95**
_____ *LADS Disk* (Atari) **$12.95**
_____ *LADS Disk* (Commodore), **$12.95**

☐ Payment Enclosed (check or money order)
☐ Charge ☐ MasterCard ☐ Visa ☐ American Express

Acct. No _____

Signature _____

Name _____

Address _____

City _____

State _____ Zip _____

_____ 1 Book for	$14.95
_____ 2 Books for	$25.00
_____ *LADS Disk* for	$12.95
NC residents add 4.5% sales tax	$_____
Shipping and handling ($2.00 per book $1.00 per disk)	$_____
Total Paid	$_____

All orders must be prepaid
Please allow 4–6 weeks for delivery

COMPUTE! Publications, Inc. abc
One of the ABC Publishing Companies

Report From The Summer Consumer Electronics Show:

An Eight-Bit BONANZA

Selby Bateman, Features Editor

Forget any rumors you've heard about weakening in the 8-bit computer lines. The Summer Consumer Electronics Show revealed plenty of new software and hardware for Commodore, Atari, Apple, and IBM 8-bit machines. Also on display were scores of new software packages for the Atari ST and a growing number for the Commodore Amiga. The happy news is that both Commodore and Atari are making efforts to extend the life of their popular 8-bit computers at the same time that they're pushing the newer 16-bit models.

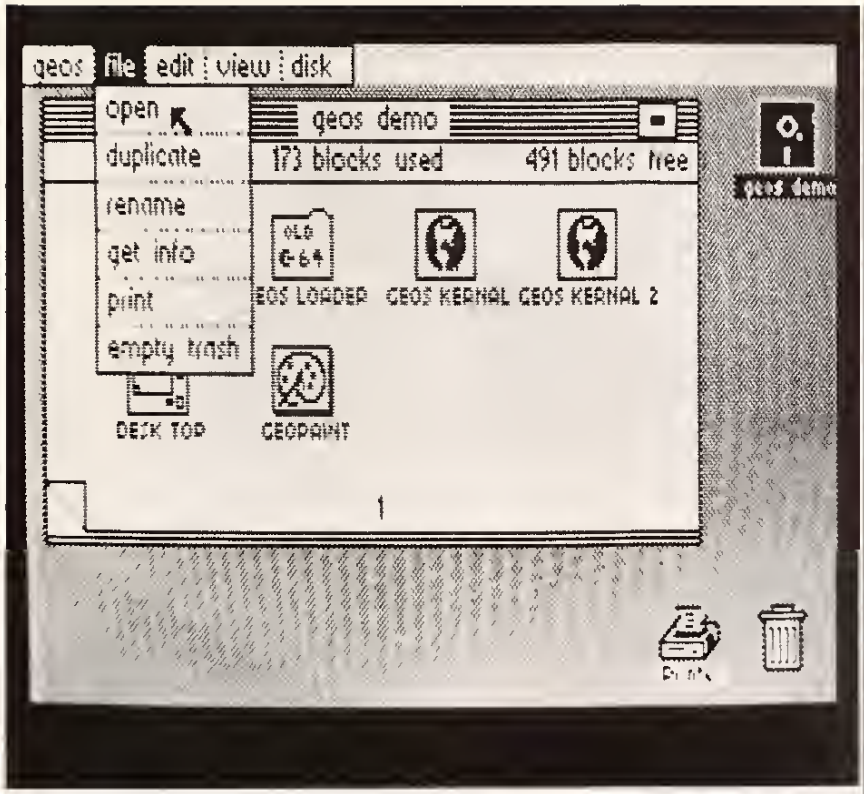
The 68000-based Amiga, ST, and Macintosh computers may be getting headlines these days, but it's the 8-bit machines which are continuing to provide much of the income for manufacturers and excitement for millions of satisfied users.

Proof of that came at the recent Consumer Electronics Show (CES) in Chicago—a semiannual showcase of all the consumer electronics products you'll be seeing on store shelves this fall and winter. Amid the newest high-tech digital audio players, 3-D televisions, videocassette machines, car stereos, credit-card-sized radios, and satellite dishes, a few dozen computer software companies displayed a wealth of new programs for Commodore, Atari, Apple, and IBM machines.

But what a difference a year can make in the fortunes of individual computer companies. Twelve months ago at CES, Commodore displayed its 64 and 128 machines in a large, heavily traveled booth on the main floor, while Atari was ensconced in a couple of meeting rooms on the mezzanine showing its fledgling Atari ST. Rumors circulated everywhere about the pending introduction of Commodore's Amiga, which was scheduled for a July release.

This year the tables were turned. While Atari occupied a large, crowded booth full of third-party software developers supporting the ST, Commodore occupied the mezzanine rooms showing its newly packaged 64. No mention was made of the Amiga, which Commodore showcased heavily at the Atlanta COMDEX show in late April, and which it obviously feels should be promoted in business markets. At a time when Atari has seen its efforts with the ST begin to bear financial fruit, Commodore has been racked by heavy financial worries. Layoffs at the West Chester, Pennsylvania, headquarters and at the Los Gatos, California, Commodore/Amiga offices occurred this spring. Sales of the Amiga have been slower than expected, and it's been the enduring strength of the 64 and the newer 128 that has helped the company fight against tremendous quarterly losses.

Even with its current financial problems, no one is counting Commodore out. In fact, the company hopes the rest of 1986 and early 1987 will see a reversal, with a leaner corporate staff, a new look for the unstoppable Commodore 64, new software and heavy sales of the 128 (now more than 600,000 sold), and a slowly rising tide of Amiga sales. Nonetheless, it's clear that the ST's popularity has hurt the Amiga. One rumor at CES, unsubstantiated at this point, is that a new, less expensive version of the Amiga is under development,



GEOS: *A new face for an old friend.*

which would compete more effectively with the ST.

The 64's New Look

As we reported in last month's "Editor's Notes," Commodore introduced at CES the new 64C, a repackaged Commodore 64 computer that cosmetically resembles the 128. Bundled with it are two disks, the first containing the icon-based GEOS operating system and *geoWrite* and *geoPaint* application programs on one side. On the other side of that disk is a terminal program for use with the Commodore-specific QuantumLink telecommunications network. The second disk contains *Odell Lake*, an educational program from MECC which teaches children about the environment within a lake. Internally, the 64C is identical to the original 64.

The 64C computer and software combo has a suggested retail price of under $250, probably around $225 according to one source. The present generation of 64s retails for about $150 nationally, but without any software. Once existing stocks of the older unit are depleted, the 64C package will be the only 64 available. The GEOS/QuantumLink disk is also available for current 64 owners for $59.95.

GEOS (Graphic Environment Operating System) brings to the 64 the type of Macintosh-style, or GEM-style, user interface available on the ST, Amiga, and Macintosh machines. GEOS loads from disk, replacing the 64's ROM-based operating system, and displays a desktop environment with icons, drop-down menus, and windows. You can use your joystick or a mouse to move around the screen. What's more, disk operations are speeded up by a factor of from five to seven times. Menu titles such as File, Edit, View, and Disk open to reveal additional choices under each heading. Also included on the disk are powerful programs for productivity applications in the home market—word processing, calculation, and graphics design. Although there are still some memory constraints imposed by GEOS on the 64's available RAM (Random Access Memory), Commodore plans to introduce later this year a memory expansion cartridge for the 64 like the unit now available for the 128. (For more information on GEOS, see the CES report in the April 1986 COMPUTE!.)

New 128-style cases have also been developed for the 1541 drive (now the 1541C) and the 1702 color monitor (now the 1802). Commodore was also displaying a new color monitor for the 64 and 128, the 1902A, which can handle composite video as well as digital RGB (red-green-blue) signals. There's also a button that gives you a green screen.

Although reactions to GEOS from software companies were mixed, the overall response seems to have been favorable, according to representatives from several companies who attended a GEOS development seminar hosted by Commodore and Berkeley Softworks. The result, if all goes according to plan, is for third-party software developers to produce programs for the new 64C that operate under the easy-to-use GEOS interface. The procedure is not difficult, says one of the manufacturers, and could provide an entirely new uni-

The new Commodore 64C sports a sleeker look. A disk containing GEOS *and Quantum Link terminal software is bundled with the computer.*

verse of software for the popular 64.

Commodore also announced that the Commodore 128 has already sold more than 600,000 units. With that installed base of machines, plus the millions of 64s already in use, Commodore feels that the computer can have a life of at least two or three more years. That's a small miracle considering the pace at which computers become obsolete. After all, the 64 is now more than four years old. But, there are more than five million of the machines out there, with a sizable number of them still in regular use.

In cooperation with the games division at Lucasfilm, Commodore demonstrated a unique new addition to QuantumLink, an online news and information service heavily supported by Commodore. *Habitat* is an interactive online activity, something of a cross between a game and the normal CB-type activity found on QuantumLink. Once you've entered the area online, you are allowed to create a graphic representation of yourself using a character construction set. Then you can explore the thousands of locales created by the Lucasfilm game staff, interacting with other people as you move around. Commodore expects this feature to be available in late summer or early fall. At press time, the hourly online charge was still uncertain.

Atari's XE Bundles

Though much of the excitement over Atari at the show centered on new products for the ST computers, Atari used a section of its large booth at CES to promote the low-cost 65XE and 130XE computers in a variety of bundled systems. A complete starter package includes the CPU, printer, disk drive, and five software titles: *AtariWriter Plus, Home Filing Manager, Music Composer, Defender,* and *Star Raiders.* The 64XE (64K) starter package retails for $349.95, and the 130XE (128K) for $399.

Atari also introduced new software titles and peripherals for the XE line. *Atari Planetarium* is an educational program that simulates a complete observatory. It can show the location of more than 1200 stars, 88 constellations, more than 300 deep-sky objects, and the path of Halley's Comet during its most recent appearance. The program retails for $24.95. *Star Raiders II* is an arcade-style game, a sequel to the 1981 *Star Raiders.* It retails for $19.95. Atari's new dot-matrix printer for the XE line, the XMM801, supports Epson medium-resolution graphics. With up to 80 characters per second, the new printer requires no special interface for the Atari XE. It supports both friction and tractor feed, and retails for $219. The XEP80, a new 80-column adapter compatible with all Atari eight-bit computers, allows for 80-column output to a standard monochrome composite monitor; it will be equipped to let the user connect a standard Centronics parallel printer. No price was available at press time.

Apple Computer, which traditionally does not exhibit at CES, was a strong presence nonetheless, as a variety of new Apple-related products were introduced by third-party software vendors. Many of those software producers were speculating on the soon-to-be-

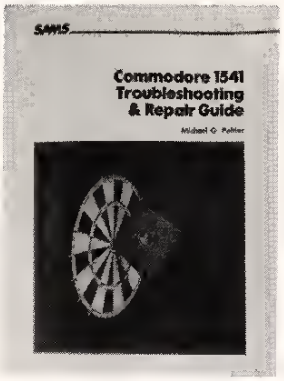
Easy microcomputer troubleshooting and repair

IBM ® PC Troubleshooting and Repair Guide
Robert C. Brenner
Even the computer novice will feel comfortable with this fully illustrated book on IBM PC troubleshooting and repair. Simple instructions help you identify the problem and tell you how to fix it quickly with few or no tools.
(No. 22358, $19.95)

Apple® II Plus/IIe Troubleshooting and Repair Guide
Robert C. Brenner
The Apple II Plus/IIe Troubleshooting and Repair Guide is complete with illustrations and photographs to guide you through the repair of your Apple II Plus or IIe microcomputer. Also included are easy to understand circuit diagrams, schematics and block diagrams. (No. 22353, $19.95)

Commodore 64™ Troubleshooting and Repair Guide
Robert C. Brenner
Keep your Commodore 64 in top operating condition with the Commodore 64 Troubleshooting and Repair Guide. Step-by-step instructions will guide you through the complexities of making simple repairs. (No. 22363, $19.95)

Commodore 1541 Troubleshooting and Repair Guide
Michael G. Peltier
If you own or operate a Commodore 64 or VIC 20 computer and are using the 1541 disk drive, this is the comprehensive servicing guide you'll need for equipment maintenance. Disassembly/reassembly instructions, theory of operation, diagrams and schematics make repair easy.
(No. 22470, $19.95)



To order call **800-428-SAMS**
In Indiana call 317-298-5566
Ask for Operator **834**

Name _____
Address _____
City _____
State _____ Zip _____

Check or money order enclosed. Make checks payable to Howard W. Sams & Co. Mail this form with payment to Howard W. Sams & Co., Dept. DM • 4300 West 62nd Street • Indianapolis, IN 46268

DM834

☐ Bill my credit card
☐ VISA ☐ MC ☐ AE Exp. Date
Account No.
Signature _____ (required)

Book No.	Quantity	Price
Shipping & Handling		$ 2.50
AR CA FL IN NC NY OH TN WV residents add local sales tax		$
Total		$

SAMS™

Earth will be destroyed in 12 minutes to make way for a hyperspace bypass.

Should you hitchhike into the next galaxy? Or stay and drink beer?

Slip the disk in your computer and suddenly you are Arthur Dent, the dubious hero of THE HITCHHIKER'S GUIDE TO THE GALAXY,™ a side-splitting masterwork of interactive fiction by novelist Douglas Adams and Infocom's Steve Meretzky. And every decision you make will shape the story's outcome. Suppose for instance you decide to linger in the pub. You simply type, in plain English:

>DRINK THE BEER

And the story responds:

YOU GET DRUNK AND HAVE A TERRIFIC TIME FOR TWELVE MINUTES, ARE THE LIFE AND SOUL OF THE PUB, THEY ALL CLAP YOU ON THE BACK AND TELL YOU WHAT A GREAT CHAP YOU ARE AND THEN THE EARTH GETS UNEXPECTEDLY DEMOLISHED. YOU WAKE UP WITH A HANGOVER WHICH LASTS FOR ALL ETERNITY. YOU HAVE DIED.

Suppose, on the other hand, you decide to:

>EXIT THE VILLAGE PUB THEN GO NORTH

In that case you'll be off on the most mind-bogglingly hilarious adventure any earthling ever had.

You communicate – and the story responds – in full sentences. So at every turn, you have literally thousands of alternatives. If you decide it might be wise, for instance, to wrap a towel around your head, just say so:

>WRAP THE TOWEL AROUND MY HEAD

And the story responds:

THE RAVENOUS BUGBLATTER BEAST OF TRAAL IS COMPLETELY BEWILDERED. IT IS SO DIM IT THINKS IF YOU CAN'T SEE IT, IT CAN'T SEE YOU.

Simply staying alive from one zany situation to the next will require every proton of puzzle solving prowess your mere mortal mind can muster. So put down that beer and hitchhike down to your local software store today. Before they put that bypass in.

Comes complete with Peril Sensitive Sunglasses, a Microscopic Space Fleet, a DON'T PANIC Button, a package of Multipurpose Fluff and orders for the destruction of your home and planet.

Other interactive science fiction stories from Infocom

INFOCOM™

For more information call 1-800-262-6868. Or write to us at 125 CambridgePark Drive, Cambridge, MA 02140.

© 1985 Infocom, Inc. THE HITCHHIKER'S GUIDE TO THE GALAXY is a trademark of Douglas Adams. PLANETFALL, STARCROSS and SUSPENDED are registered trademarks. A MIND FOREVER VOYAGING is a trademark of Infocom, Inc.

announced Apple II 16-bit computer.

Another popular topic of industry conversation centered on the swiftly dropping prices of IBM PC work-alikes, called clones, that are expected to be as low as $300 by the Christmas season. The IBM clones, from Korea, Taiwan, Japan, and even the U.S., are already beginning to sell into consumer markets. And that trend is expected to continue. Heavy sales of the Tandy 1000 and rumors about extremely inexpensive clones have caused some software publishers to consider beefing up their IBM offerings.

Although a complete list of software and hardware showcased at CES is beyond the scope of this article, the following products were among those introduced for Apple, Atari, Commodore, and Atari 8-bit computers. For more product information, see the "News & Products" section in this issue; for information on new products introduced for the 16-bit machines, see Tom Halfhill's story elsewhere in this issue.

Electronic Arts: Electronic Arts continues its major commitment to eight-bit computer owners with a long list of new titles for all machines. Among the new offerings are *Amnesia* (Commodore 64 version, $39.95; Apple II, $44.95), by Thomas M. Disch and Cognetics; *Autoduel* (Commodore 64, $49.95), by Origin Systems; *Bard's Tale II: The Archmage's Tale* (Commodore 64), by Michael Cranford; *Battlefront* (Commodore and Apple versions, $39.95), by Strategic Studies Group; *Chessmaster 2000* (Commodore, Apple, and Atari versions, $39.95; IBM, $44.95), by Software Country; *Scavenger Hunt* (Commodore and Apple II), by Ozark Softscape; *Timothy Leary's Mind Mirror* (Commodore version, $32.95; Apple II, $34.95), by Dr. Timothy Leary; *Ultimate Wizard* (Commodore 64, $29.95), by Sean A. Moore and Steven Luedders; *Age of Adventure* (Apple II and Atari, $14.95); and *Venture's Business Simulator* (IBM only, $99.95), by Reality Development. (Electronic Arts, 1820 Gateway Dr., San Mateo, CA 94404.)

Abacus Software: In addition to its line of Atari ST and Commodore 128 books, Abacus displayed its *BASIC Compiler for the 128* ($59.95) along with the previously released 64 version ($39.95). Also on display were the 128 versions of its CADPAK computer-aided design program, *Super-C Language Compiler* and *Super Pascal Development System* ($59.95 each; 64 versions also available). Among a variety of other software packages, Abacus has now added *COBOL-64*, a Commodore version of the popular business programming language. (Abacus Software, P.O. Box 7219, Grand Rapids, MI 49510.)

The Learning Company: Two new products have been added to its collection of well-known educational software. *Writer Rabbit* helps develop the critical process of learning to use words and sentences. It offers several features that were implemented in response to requests made by children, teachers, and parents. The program includes several games, each of which enables the child to explore a different aspect of words and sentences in a fun and supportive setting. The games incorporate graphics and sound, and each game can be tailored to a child's own pace.

Math Rabbit teaches early math skills to children ages 5-7, and also incorporates entertainment to encourage children to participate. Available for Apple II series computers, each program has a suggested retail price of $39.95. (The Learning Company, 545 Middlefield Rd., Suite 170, Menlo Park, CA 94025.)

Access Software: On the heels of its popular *Leader Board* professional golf simulator, Access introduced *10th Frame* ($39.95), a professional bowling simulator for the Commodore 64. (Access Software, 2561 South 1560 West, Woods Cross, UT 84087.)

Multibotics: In cooperation with Access Software, this company is introducing a line of home robotics workshops for the Commodore 64 and 128, Atari 400/800/XL/XE, Apple IIe, IBM PC and compatibles, Commodore Amiga, and Atari ST.

The MB230 Workshops consist of an interface that connects the computer to snap-together robotics modules, plus software for controlling the modules. The software enables the computer to function as a variable-speed motor controller, a voltmeter, an oscilloscope, an infrared controller/detector, and an audio digitizer. Retail prices for the workshops range from $59.95 to $199.95. (Access Software, see address above.)

Accolade Software: Accolade is introducing in late summer an arcade-action game called *Deceptor.* As you manipulate your Deceptor through six levels of increasingly difficult play, you can transform the robotic vehicle from ground-based to airborne, and finally into a humanoid shape. The game's responsiveness can be tailored to your liking, and you can practice most of the levels to increase your chances of survival. (Price unannounced.)

Accolade also announced Apple II and IBM versions ($34.95 each) of its *PSI-5 Trading Company* science fiction adventure game. A Macintosh version ($44.95) of the *Hardball* baseball game was also announced at CES. (Accolade Software, 20833 Stevens Creek Blvd., Cupertino, CA 95014.)

Springboard Software: The publisher of the bestselling *Newsroom* has introduced two new productivity packages with application in the home, school, and office.

The Newsroom Pro is aimed at the person who wants to take a more professional approach to producing a newsletter. It contains everything the user needs to produce a high-quality newsletter, including banner creation, text entry, graphic production, layout, and high-resolution printing. More than 2,000 pieces of clip art are included. It is available for the IBM-PC for $129.95.

Certificate Maker provides more than 200 predesigned certificates, awards, diplomas, and licenses in a wide variety of categories such as sports, academic achievement, families, children, religion, and business. Available for Apple ($49.95), IBM-PC ($59.95), and Commodore 64 (price not yet determined). (Springboard Software, 7808 Creekridge Cir., Minneapolis, MN 55435.)

Activision: The Activision family of companies continues to expand, with the acquisition of Infocom, a well-known adventure game company. Infocom will maintain its own brand-name imprint under the Activision umbrella. Pre-

COMMODORE 64 BESTSELLERS

from COMPUTE! Books

You can depend on COMPUTE! for clearly written, easy-to-use books for your Commodore 64. This assortment of titles includes some of our most frequently requested books containing many of our best ever applications, games, utilities, tutorials, and programming hints, all ready to type in and use on your Commodore 64.

Look over this list of backlist bestsellers, and choose the titles you need to complete your library of first-rate Commodore 64 books from COMPUTE!.

COMPUTE!'s First Book of Commodore 64 Sound and Graphics

Edited, 275 pages
Clear, useful explanations of the 64's sound and graphics capabilities including tutorials and example programs.
$12.95 ISBN 0-942386-21-3

Creating Arcade Games on the Commodore 64

Robert Camp, 357 pages
A guide to creating arcade games on the 64, plus finished games to play and study.
$14.95 ISBN 0-942386-36-1

COMPUTE!'s Commodore Collection, Volume One

Edited, 208 pages
An anthology of 28 practical programs in easy-to-use form for the Commodore 64 and VIC-20.
$12.95 ISBN 0-942386-55-8

COMPUTE!'s Machine Language Routines for the Commodore 64

Edited, 255 pages
Complete machine language programs and easy-to-use routines help to make the Commodore 64 more powerful and versatile.
$14.95 ISBN 0-942386-48-5

COMPUTE!'s Commodore Collection, Volume Two

Edited, 270 pages
This second volume in COMPUTE!'s Commodore Collection series includes exciting games, sophisticated applications, versatile educational routines, and helpful programming aids for VIC-20 and Commodore 64 users.
$12.95 ISBN 0-942386-70-1

All About the Commodore 64, Vol. One

Craig Chamberlain, 289 pages
For beginning to intermediate programmers who want to develop the full potential of their Commodore 64 computers.
$12.95 ISBN 0-942386-40-X

Visit your local book or computer store for these titles. Or order directly from COMPUTE!. To order, call toll-free 800-346-6767 (in NY 212-887-8525) or write COMPUTE! Books, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

Please allow 4–6 weeks for delivery.

When ordering, please include $2.00 shipping and handling per book in U.S. and surface mail or $5.00 airmail. North Carolina residents add 4.5 percent sales tax. U.S. funds only.

COMPUTE! Publications, Inc. abc
Part of ABC Consumer Magazines Inc
One of the ABC Publishing Companies
825 7th Avenue 6th Floor, New York NY 10019
Publishers of COMPUTE! COMPUTE!'s Gazette COMPUTE!'s Gazette Disk COMPUTE! Books, and COMPUTE!'s Apple Applications

COMPUTE! books are available in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England, and in Canada from McGraw-Hill, Ryerson Ltd., 330 Progress Ave., Scarborough, Ontario, Canada M1P 2Z5.

viously acquired companies, such as Creative Software and Gamestar, continue to have an impact on the company's product line as well.

I Am the C-128 is one of the products in Activision's new Personal Choice Software line, which includes the *Writer's Choice* word processor, the *Filer's Choice* database, and *Planner's Choice* spreadsheet for the Apple II family, the Commodore 64/128, and the IBM PC, the Tandy 1000, and other MS-DOS computers. One of Activision's most popular products last year was the mystery adventure game *Hacker*. This year the company will introduce a sequel, *Hacker II: The Doomsday Papers*, which begins where the first program ended. Commodore 64/128 and Apple II versions will sell for $39.95, while IBM PC/PCjr/Tandy 1000 and Macintosh versions will be priced at $49.95 each. Activision remains one of the most prolific software publishers, with more titles scheduled for release after September 1. (Activision, 2350 Bayshore Frontage Road, Mountain View, CA 94043.)

Simon & Schuster: The software division of this publishing house has released *Webster's New World Writer*, a versatile word processor (IBM-PC with 256K, $150), and *Webster's New World On-Line Thesaurus*, a 120,000-word thesaurus compatible with more than 30 major word processors and other software packages (IBM-PC with 128K, PCjr with 256K; $69.95). (Simon & Schuster Software, Gulf & Western Bldg., One Gulf & Western Plaza, New York, NY 10023.)

Avalon Hill: The Microcomputer Games division of Avalon Hill also announced a variety of new titles for Apple, Atari, Commodore, and IBM computers.

Spitfire 40 is a World War II air-war game and flight simulator for the Commodore 64/128 ($35), with conversions for other machines already under way. The popularity of Avalon Hill's *Super Sunday* football game has encouraged the company to introduce 1985 expansion disks for use with the original game, for Commodore 64/128 and IBM machines ($20 each).

Macbeth is a graphics-and-text adventure game based on the Shakespearean play, for Commodore 64/128 ($25). In August, Avalon Hill will introduce *Darkhorn*, a fantasy warfare game, for the Apple II and Commodore computers ($30). A science fiction arcade-action game, *Mission on Thunderhead*, is now available for Apple II, Atari 800/XL/XE, and Commodore 64/128 computers ($25). Expansion modules, one for Extended Units and the other for the Campaign Disk, are also available for the previously released *Under Fire!* strategy game. (Avalon Hill, Microcomputer Games Division, 4517 Harford Road, Baltimore, MD 21214.)

Bantam Electronic Publishing: Two new Apple II and Commodore 64 programs scheduled for fall release were displayed by Bantam at CES. The packages feature popular Disney cartoon characters in productivity programs.

Each program will carry a retail price of $39.95 for Apple II versions, and $34.95 for Commodore 64/128 versions. (Bantam Electronic Publishing, 666 Fifth Ave., New York, NY 10103.)

Softsync: This company has premiered *The Model Diet* (Commodore 64, $29.95; Apple II, IBM-PC, $34.95), a computerized diet and nutrition program; and *Desk Manager* (Commodore 64, 128, Apple II, $39.95), a desktop accessory that uses windows. (Softsync, Inc., 162 Madison Ave., New York, NY 10016.)

Batteries Included: As noted in last month's "News & Products" (page 117), Batteries Included has introduced an extensive array of new products for a variety of computers. Among the new titles you'll be seeing will be the *PaperClip II* word processor ($79.95) and the *HomePak* three-in-one telecommunications–word processor–data manager ($49.95), both for the Commodore 128; the *PaperClip* word processor for the Apple II/II+/IIe/IIc computers ($59.95); *PaperClip with SpellPak* spelling checker for the Atari 130XE ($59.95); and five new productivity packages for the IBM PC and compatibles, including the advanced *PaperClip Elite* word processor ($129.95) and *Degas Elite* graphics program ($79.95), among others. (Batteries Included, 30 Mural St., Richmond Hill, Ontario, Canada L4B 1B5.)

Spinnaker: This software publisher introduced *A.C.E.*, a combat simulator for the Commodore 64. This game combines a flight simulator with arcade-game-style combat. It features multiple weapons systems, an on-board computer, overhead satellite mapping, and 3-D action ($19.95). (Spinnaker Software, One Kendall Sq., Cambridge, MA 02139.)

Bodylog: Bodylog has developed a new multipurpose peripheral called Bodylink, which plugs into the cartridge slot of a Commodore 64/128 and turns the computer into an exercise machine, stress reduction device, and personal computerized biofeedback loop. Once you've purchased a package that contains the main Bodylink hardware, you can buy add-on software packages for whatever applications you're interested in. Prices for starter kits range from $139.95 to $209.95; additional hardware and software packages for a wide variety of applications cost between $29.95 and $99.95. (Bodylog, 34 Maple Ave., Armonk, NY 10504.)

Timeworks: Timeworks announced that its Commodore 128-specific programs, *Word Writer 128*, *Swiftcalc 128*, *Data Manager 128*, and *Sylvia Porter's Personal Financial Planner 128*, will continue to be upgraded on a regular basis. The publisher has also added a thesaurus to *Word Writer 128*. (Timeworks, 444 Lake Cook Rd., Deerfield, IL 60015.)

Brøderbund Software: Several new products representing a diverse line of software were introduced by Brøderbund. Among them were *The Toy Shop*, available for the Apple II series and Commodore 64, which lets the user make 20 working mechanical models and toys. Users can customize their toys, print out the designs on paper, and attach them to adhesive cardboard. Wire, wooden dowels, adhesive cardboard, and other necessary supplies are included in the package, along with a comprehensive user manual. Suggested retail price for both versions is $59.95. (Brøderbund Software, 17 Paul Dr., San Rafael, CA 94903.)

For further information on new products announced at the Summer Consumer Electronics Show, please see the "News & Products" section. ©

16-Bit Explosion!

New Products For The Atari ST And Amiga

Tom R. Halfhill, Editor

As they enter their second year on the market, the Atari ST and Commodore Amiga are building up respectable software libraries spanning all the major categories of personal computing. At the same time, new peripherals and accessories are making the computers themselves even more powerful. Here's a look at the highlights of two recent computer industry trade shows: the Spring COMDEX in Atlanta and the Summer Consumer Electronics Show (CES) in Chicago. Many of these new products will be available this summer.

Atari ST

Atari was a major player at the Spring COMDEX and Summer Consumer Electronics Show (CES), filling its booths at both shows with dozens of cubicles sponsored by independent developers demonstrating their wares. The exhibits attracted thousands of browsers and potential new dealers. Perhaps more importantly, Atari continued to gain credibility—strengthening its image as a revitalized company on firm financial footing which is determined to become a significant force in the personal computer industry.

Atari's biggest announcements for the ST series included:

- An MS-DOS emulator that is supposed to run most of the big-name IBM PC software. (The prototype was running Microsoft's *Multiplan*.) The emulator is an external box which contains an 8088 microprocessor, a socket for an 8087 math coprocessor, and 512K of random access memory (RAM). When the emulator isn't operating, the ST can use the extra 512K as a RAM disk. Atari still hasn't decided whether to put a 5¼-inch floppy disk drive in the box, so the final price is undetermined. Estimates are $300 to $400. Atari plans to begin selling the emulator this fall.
- A CP/M emulator implemented entirely in software. This comes on a 3½-inch disk and lets you run virtually any program written for the CP/M (Control Program/Microcomputers) operating system at 100 percent speed. No extra hardware is required. Already available in Europe, the CP/M emulator should be selling in the U.S. this summer for under $50.
- A special summer price promotion that allows dealers to sell a 520ST, floppy disk drive, and monochrome monitor for $599.
- Atari announced immediate availability of its 20-megabyte SH204 hard disk drive for $799.95 and an Epson-compatible dot-matrix printer, the SMM804, for $219.95. The printer can make accurate screen dumps of the ST's high-resolution (640 × 400-pixel) screen mode. It prints at 80 characters per second and offers both friction and tractor feed.
- Atari has acquired rights to market an ST version of Versasoft's *dBMAN*, a high-end relational database manager originally designed for the IBM PC and patterned after Ashton-Tate's *dBASE II* and *dBASE III*. According to Atari, experienced *dBASE* users can use *dBMAN* with no retraining. The suggested retail price is $149.95, and Atari is encouraging dealers to give free evaluation copies to potential customers. The free copy is fully functional, but allows only 30 records per database.

In addition to these announcements, independent companies exhibited a flood of new software and hardware for the ST series, including some impressive business programs. With even more products due this fall, it's obvious that the ST will have a solid software library by the end of 1987.

So much software is being released that we don't have room here to cover it all; scan the "News & Products" section in this and future issues for further information. Following are some products that particularly caught our attention.

Batteries Included (Irvine, California) is emerging as one of the top software companies supporting the ST. Later this summer it plans

$10,000.00

Atari ST Programming Contest!

First Prize $5,000.00 **Second Prize $2,500.00**

Third Prize $1,000.00

Three Honorable Mentions $500.00 each

COMPUTE! Publications, Inc. is looking for the very best original software for the Atari ST series computers. And to prove we're serious, we're offering a total of $10,000.00 in prize money to the top six winners. That's $5,000.00 for First Prize, $2,500.00 for Second Prize, $1,000.00 for Third Prize, and $500.00 each for three Honorable Mentions. In addition, the winners will receive our standard royalties when their programs are published. And even if your program doesn't win a prize, you can still earn purchase fees and royalties if we accept your entry for publication.

Interested? If so, read these rules:

1. Entries must be your original work, previously unpublished. All those whose programs are accepted will be required to affirm this in writing.

2. You can submit as many entries as you want, but we cannot consider programs which have been entered in other contests or submitted for publication elsewhere at the same time.

3. The deadline is October 1, 1986. All entries must be received at our offices by this date. Programs submitted after this date will still be considered for publication, but will not be entered in the contest.

4. Entries are allowed (and encouraged) in virtually all software categories: home and business applications, education, recreation, telecommunications, graphics, sound and music, utilities, and desk accessories.

5. Entries may be written in any programming language—including BASIC, Logo, C, machine language, Pascal, Modula-2, Forth, FORTRAN, and Prolog—as long as they meet two requirements. First, if you're using a compiled language, the compiled object or run-time code must be a self-standing program that can be run by someone who doesn't own a copy of the language. (Exceptions are ST BASIC and Logo. Since these languages come with the ST, it can be assumed that everyone owns a copy.) Second, we must be able to legally distribute the program without incurring licensing fees or other obligations to the maker of the language. If you're not sure whether a certain language qualifies, contact its maker for clarification.

6. Entries must be submitted on a single- or double-sided 3½-inch ST disk with both the run-time code and source code included.

7. Entries must be accompanied by an article which explains how to use the program, what it does, and so on. If your program employs any new or unusual techniques that you think will be of interest to other ST programmers, you can also describe how the program works.

8. Submissions which do not win a prize and are not accepted for publication will be returned only if accompanied by a self-addressed, stamped mailer.

9. All judging will be handled by the staff of COMPUTE! Publications, Inc. All decisions regarding contest entries and acceptances will be solely at the discretion of COMPUTE! Publications, Inc., and all decisions are final. This includes decisions regarding creativity, similarity among entries, and so forth.

10. Winners will be announced by COMPUTE! Publications, Inc. in late 1986.

11. This contest is void where prohibited by law. Full-time, part-time & previous employees of COMPUTE! Publications, Inc., and Capital Cities/American Broadcasting Corporation are ineligible for the contest, but may still submit work for publication at standard rates.

Every Contest Entry Must Contain This Form:

I warrant that the program presently entitled_____

_____ is my own original work and that the work has not been submitted for consideration elsewhere, nor has it been previously published. If my work is accepted by you, I understand that your decision as to the selection of winners and awarding of prizes is final and without recourse on my part. I agree, should you select my submission, to sign your standard contract, which includes assignment of the copyright of the program to COMPUTE!, and to allow you to use my name and image in promotional materials and other forms. (*If you are under age eighteen, your parent or legal guardian must sign for you.*)

Address entries to:

ATARI ST CONTEST
COMPUTE! Publications, Inc.
P.O. Box 5406
Greensboro, NC 27403

COMPUTE! Publications, Inc.

Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies

Announcing major news for Atari ST users:

Special BONUS First Issue FREE with Subscription

COMPUTE!'s Atari ST DISK & MAGAZINE

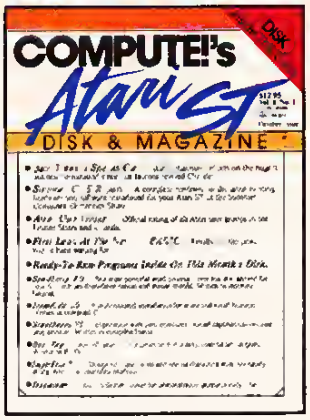
A bimonthly magazine devoted exclusively to Atari ST enthusiasts that includes a disk containing all of the programs found in each issue.

Atari has proven the pessimists wrong. The Atari 520ST and 1040ST have become the bestsellers among the new generation of personal computers. Both are breakthroughs in price and performance, and the community of ST owners is growing by thousands each month.

That's one reason why COMPUTE! Publications is announcing a new magazine specially designed for ST users. At the same time, we recognize that the power of the ST presents a unique challenge to magazines which publish program listings. That's why we're including a 3½-inch disk that contains every program found in each issue—ready to load and run. No more typing!

Here's what you'll get in every issue of **COMPUTE!'s Atari ST Disk & Magazine:**

- **Top-quality programs.** Utilities. Games. Educational programs for youngsters. Application programs for home and business. And since all the programs will be on disk, there are few limitations on length or languages. A typical disk might contain an elaborate adventure game written in BASIC, a programming utility written in machine language, a dazzling graphics demo in compiled Pascal, and a useful home or business application written in Forth or C.
- **Neochrome of the Month.** Take a look at what computer artists are doing with the Atari ST. Each issue's disk contains a *Neochrome* picture file ready for you to load and admire. Are you an artist yourself? Send us a picture of your own, and we'll pay you if it's published.
- **Regular columns.** If you're a programmer—or would like to be—you'll love our columns on ST programming techniques and the C language. Or check out our column on the latest events and happenings throughout the ST community. Or send your questions and helpful hints to our Reader's Feedback column.
- **Reviews.** Honest evaluations of the latest software and hardware for the Atari ST.
- **News & Products.** A comprehensive listing of the newest releases for your ST.



- **And more:** Interviews with ST newsmakers, reports on the latest industry trade shows, and overviews of significant new product introductions.

Starting with the October issue (available September 1), **COMPUTE!'s Atari ST Disk & Magazine** will be found on newsstands nationwide for only $12.95 per copy, including disk. Or it can be delivered directly to your mailbox six times a year for only $59.95—a savings of over 20 percent.

As a special bonus, if you order a prepaid subscription before August 1, you'll get the first issue absolutely free!

To order, call 800-346-6767. In NY 212-887-8525
or send check or money order to
COMPUTE!'s Atari ST Disk & Magazine
ABC Consumer Magazines, Inc.
Circulation Dept./8th Floor
825 7th Avenue
New York, NY 10019

COMPUTE! Publications, Inc. abc
Part of ABC Consumer Magazines Inc
One of the ABC Publishing Companies

Atari is a trademark of Atari Corporation

to release a follow-up to its popular *Degas* drawing program: *Degas Elite*. New features include ten levels of magnification; the ability to load a picture created in any resolution into any other screen mode (including monochrome to color and vice versa); the ability to load pictures created with an Atari 400/800/XL/XE and KoalaPad or Atari Touch Tablet; up to eight screens in memory at once, with block-copying between screens; adjustable color cycling for animation effects; automatic color blending across the selected color palette; and the ability to grab any portion of a screen and use it as a paintbrush. *Degas Elite* will sell for $79.95.

Batteries Included has already started shipping a program called *Thunder!*, a realtime spelling checker. *Thunder!* installs as a desk accessory and loads a 50,000-word dictionary into memory and, using a special compaction technique, takes up only about 80K of RAM. It works in realtime with any program that supports GEM—including word processors, terminal programs, text editors, and notepads. When you type a word that *Thunder!* cannot find it its dictionary, it beeps to let you know. By pressing a key or selecting a menu item, you can pop open a window that displays a number of words that *Thunder!* thinks you were trying to spell. If you find the correct word in the list and click on it with the mouse, *Thunder!* automatically substitutes the correct spelling, closes the window, and lets you resume typing. If you find realtime spell-checking annoying, *Thunder!* also lets you check an entire document after it's created or check documents created with text editors that don't support GEM. Numerous other features allow you to add your own words to the main dictionary, compile supplementary dictionaries on disk, and analyze your text for readability. *Thunder!* sells for $39.95.

Abacus Software (Grand Rapids, Michigan) announced several new programs: *ST TextPro*, a word processor with mouse and keyboard commands, multicolumn and sideways printing, user-definable function keys, automatic indexing, and table-of-contents generation; *ST Text Designer*, a page-making package for creating layouts from text files; *ST DataPro*, a database manager that allows up to 64,000 records of unlimited length; *ST Forth/MT*, a multitasking Forth with more than 1500 commands and 32-bit arithmetic; *ST PaintPro*, a GEM-based design program; and *ST AssemPro*, a 68000 macro assembler and debugger with text editor. All these programs sell for $49.95, except *ST AssemPro*, which sells for $59.95.

The software company which wrote *1st Word* for Atari—GST of Cambridge, England—is exporting several programs to the U.S., including *1st Word Plus*. Among other things, this enhanced word processor lets you merge *Neochrome* or *Degas* pictures into documents. Current plans call for Atari to market *1st Word Plus*, but GST will be selling its other programs independently. These include *GSTC Compiler*, a GEM development package for the C language; *GST-ASM*, a 68000 macro assembler; *GEM Screen Editor*, a text editor; and *GST Linker*, for compiling runtime code from source libraries. *GEM Screen Editor* and *GST Linker* are included with *GSTC Compiler* and *GST-ASM*. Prices were not available at press time.

Avila Associates (Lafayette, California) is bringing out an animation program called *Make It Move*. By pointing and clicking on icons representing different functions, you can write a script for animating shapes, text, and other graphics. It's compatible with all of the popular drawing programs and offers such functions as zooms, fades, and spins. Price: $49.95. Another Avila product is *Casino Craps*, a complete craps simulation: $39.95.

Desk accessories are proving to be as popular on the ST as they are on the Macintosh and IBM. Two of the most complete business-oriented accessories we've seen are from Blue Moon Software (Lenexa, Kansas). *MacroDesk* contains an 18-function calculator with ten memories that works in either algebraic or reverse-Polish notation; an alarm clock/calendar that helps you keep track of events far into the future; a filer with search, print, and phone-dialing functions; and an event log that's somewhat like a diary for jotting down important contacts and events. *MacroManager* has all the features of *MacroDesk* plus a project-scheduling worksheet and a log for project time recording and analysis. *MacroDesk* sells for $39.95 and *MacroManager* for $69.95; both are available now.

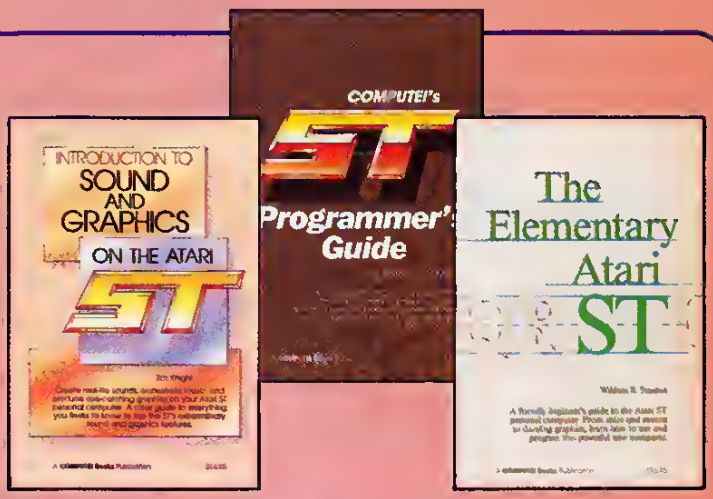
Musicians will be interested in new software from Hybrid Arts (Los Angeles). *DX-Droid* and *Oasis* take advantage of the ST's high-resolution graphics and built-in MIDI (Musical Instrument Digital Interface) ports. *DX-Droid* is a multi-featured patch editor which can even generate banks of new sounds on its own (for the Yamaha DX- and TX-series synthesizers). *Oasis* is a full-featured sampling wave-table editor and librarian for the Ensoniq Mirage. *DX-Droid* is available now for $244.44; *Oasis* should be available soon and will cost about the same as the Atari 130XE version ($187.87).

MichTron (Pontiac, Michigan) released a number of new products including *ALT*, which permits you to assign strings of up to 60 characters to each of the 36 Alternate-key combinations ($29.95); *The Animator*, a graphics-animation utility ($39.95); *BBS 2.0*, a revised version of MichTron's earlier Bulletin Board System ($79.95); *Cornerman*, a desk accessory with notepad, calculator, address book/dialer, character-code chart, clock, and a game ($49.95); *Echo*, which lets you plug in X-10 modules for controlling home appliances ($39.95); *Mighty Mail*, a mailing list manager and phone book ($49.95); and two arcade-style games, *Major Motion* and *Mission Mouse* ($39.95 each).

If you like to write your own software and want to go beyond ST BASIC and DR Logo, a few new languages are being released for the ST this summer. Softworks Limited (Chicago) is bringing out *Softworks BASIC*, a compiler that offers advanced features such as data structures like those found in C and Pascal. The XCALL statement can access machine language routines, and the TOOLBOX command lets you call most of the graphics and sound functions built into the ST's operating system. Price: $79.

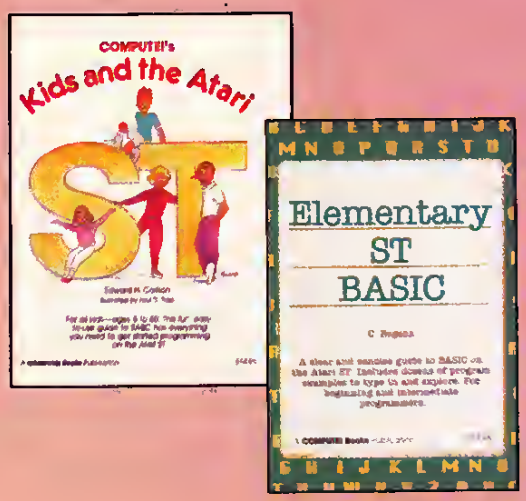
Prospero Software Limited (London) is exporting *Pro FORTRAN-77* and *Pro Pascal*, two high-

COMPUTE! Books' ATARI ST Collection



COMPUTE! Books offers you a superior line of titles for the new Atari ST. Packed full of useful utilities, exciting games, in-depth tutorials, and valuable applications, these clearly written books bring you fully tested information and entertainment for the whole family.

Look for these COMPUTE! books at your local book or computer store.



You can order directly from COMPUTE! by calling 800-346-6767 (in NY call 212-887-8525) or by sending your payment to COMPUTE! Books, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

Please include $2.00 per book shipping and handling for U.S. and surface mail or $5.00 for airmail. North Carolina residents add 4.5 percent sales tax.

Please allow 4–6 weeks for delivery from receipt of order.

COMPUTE! books are available in the U.K., Europe, the Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England, and in Canada from McGraw-Hill, Ryerson Ltd., 330 Progress Ave., Scarborough, Ontario, Canada M1P 2Z5.

COMPUTE! Publications, Inc. abc

Part of ABC Consumer Magazines, Inc
One of the ABC Publishing Companies
825 7th Avenue 6th Floor New York NY 10019

Publishers of COMPUTE! COMPUTE!'s Gazette COMPUTE!'s Gazette Disk COMPUTE! Books and COMPUTE!'s Apple Applications

COMPUTE!'s First Book of the Atari ST

Edited
A valuable collection of ready-to-type-in-and-use applications, games, and utilities. Graphics utilities like "ST Doodler," games like "Switchbox" and "Tug-a-War," and educational programs like "Hickory Dickory Dock" turn your Atari ST into everything from a business graphics machine to a powerful teaching tool. Tutorials show you how to add power to ST BASIC and how to add excitement to your own creations with sound effects. *A disk is available for $15.95 which includes all the programs in the book, 203BDSK.*
(September release)
$16.95 ISBN 0-87455-020-3

The Elementary Atari ST

William B. Sanders, 272 pages
A friendly, easy-to-use guide to the Atari ST, this book takes you through connecting your computer, loading programs, creating graphics and music, and writing your own programs.
$16.95 ISBN 0-87455-024-6

Elementary ST BASIC

C. Regena, 208 pages
A tutorial and reference guide to the ST's impressive graphics, animation, and sound with complete descriptions of ST BASIC's commands, syntax, and organization. *A disk is also available for $15.95 which includes programs from the book, 343BDSK.*
$14.95 ISBN 0-87455-034-3

COMPUTE!'s Kids and the Atari ST

Edward H. Carlson, 238 pages
Easy-to-understand instructor notes, lessons, assignments, and lively illustrations help both kids and adults painlessly learn to program on the Atari ST. The latest in the bestselling series by this author.
$14.95 ISBN 0-87455-038-6

COMPUTE!'s ST Programmer's Guide

Editors of COMPUTE!, 356 pages
A comprehensive reference guide to the Atari ST, this book explores in detail Logo and BASIC, the advanced features of the ST such as GEM and TOS, and every aspect of programming from concepts to actual program writing.
$16.95 ISBN 0-87455-023-8

Introduction to Sound and Graphics on the Atari ST

Tim Knight, 197 pages
Thorough descriptions of the Atari ST's color graphics and sound abilities, plus all the information needed to create a complete sound and graphics system.
$14.95 ISBN 0-87455-035-1

All the exciting, entertaining, and educational games, applications, and utilities from COMPUTE! magazine are now available on disk

for your Commodore, Atari, Apple, or IBM personal computer.

The COMPUTE! Disk

A new *COMPUTE! Disk* is published every month, rotating among the four major machines covered by *COMPUTE!*: Commodore 64 and 128; Atari 400/800, XL, and XE; Apple II-series; and IBM PC, PCjr, and compatibles.

Every three months you can receive a disk with all the quality programs from the previous three issues of *COMPUTE!* that will run on your brand of computer.

Like the popular *COMPUTE!'s Gazette Disk*, the *COMPUTE! Disk* is ready-to-load and error-free. It saves you valuable hours of typing time and eliminates typing errors.

With a subscription, you will receive one disk every three months for a total of four disks a year—for only $39.95. That saves you $20 a year off the single-issue cost.

Or you can order individual issues of the *Disk* for $12.95 a disk plus $2.00 shipping and handling.

Remember to specify your type of computer when ordering the *COMPUTE! Disk*. You'll find more information about this month's *COMPUTE! Disk* in this issue. (Note: You'll need the corresponding issues of *COMPUTE!* magazine to use the *Disk* since the disk will have no documentation.)

For fastest service when ordering a subscription to the *COMPUTE! Disk*, call toll free 1-800-247-5470 (in Iowa 1-800-532-1272).

For more details or to order individual issues of the *COMPUTE! Disk*, call our Customer Service Department toll free at 1-800-346-6767 (in New York 212-887-8525).

Please allow 4–6 weeks after placing an order for your first disk to arrive.

COMPUTE! Publications, Inc. abc
One of the ABC Publishing Companies
825 7th Avenue 6th Floor New York NY 10019
Publishers of COMPUTE! COMPUTE!'s Gazette COMPUTE!'s Gazette Disk COMPUTE Books and COMPUTE's Apple Applications

level compilers. Both languages have 7- and 16-digit precision floating-point math, four-byte integers, and the ability to access GEM routines. Each costs $149. (The U.S. distributor is Apex Resources, Brookline, Massachusetts.)

TDI Software (Dallas) has released two new versions of its Modula-2 compiler, including a special developer's version with directory search paths, a symbolic debugger, new modules, an intelligent linker, an enhanced text editor, and improved documentation on GEM. The regular version is $79.95, and the developer's version is $149.95. Upgrades for current owners are available at less cost.

Several companies are releasing significant small-business software for the ST. Timeworks (Deerfield, Illinois) is introducing *Word Writer ST*, a word processor with an 85,000-word spelling checker and thesaurus, outlining, macro keys, and GEM interface; *SwiftCalc ST*, a spreadsheet program which can translate data into pie charts, bar charts, scatter diagrams, line graphs, and 3-D staggered bar charts, plus sideways printing for wide spreadsheets; and *Data Manager ST*, a database manager with graphics and functions for generating labels and reports. All three programs are integrated with each other and sell for $89.95 each.

Sierra On-Line (Mountain View, California) is releasing a small-business accounting package called *ST OneWrite*. It automatically posts ledgers and prints out checks on standard business forms. Price: $129.95. Oxxi (Long Beach, California) is introducing *dbOne*, a database manager that is compatible with *dBASE II* files. Price: $99. And Dac Software (Dallas) is translating two of its popular IBM PC packages for the ST: *Dac-Easy Accounting* ($69.95) and *Dac-Easy Payroll* ($49.95).

A variety of games are coming out for the ST this summer, and although many are translations from versions previously available on other computers, some are brand-new.

Activision (Mountain View, California) is introducing *Hacker II: The Doomsday Papers*, a sequel to the popular *Hacker* ($49.95), and *The Activision Little Computer People Discovery Kit*, which simulates living creatures inside your computer. *Little Computer People* is already available on other machines. Another Activision product—which isn't a game—is *Paintworks*, a graphics-design program. (Originally known as *N-Vision*, *Paintworks* was written for Activision by Audio Light.) One feature that sets *Paintworks* apart from all other drawing programs on the ST is that you can design a picture taller than the screen—as large as an 8½ × 11-inch page, in fact. You can scroll the picture vertically and make a full-size hardcopy with an appropriate color printer, such as the Okimate 20. Price: $69.95.

The Avalon Hill Game Company (Baltimore) is releasing *Spitfire 40*, an authentic flight simulator that puts you in the cockpit of a Royal Air Force fighter plane during the Battle of Britain. It even recreates the fuel pump problems experienced by Mark 1 Spitfires while diving. Price: $35. Avalon Hill also is working on a football simulation due for release later this year.

Cosmi (Wilmington, California) is completely rewriting its *Super Huey Helicopter Flight Simulator* for the ST to take advantage of the computer's enhanced graphics. Price: $39.95. And Microprose (Hunt Valley, Maryland) is doing likewise with *Silent Service*, its much-praised World War II submarine simulation. Microprose also hinted that two more of its simulations will be rewritten for the ST later this year.

Infocom (Cambridge, Massachusetts), which recently merged with Activision, introduced a few new works of text-only interactive fiction for $39.95 each. (They're also available for the Amiga and several other machines.) *Trinity* places you in London just as World War III begins. As The Bomb begins exploding overhead, you enter a mysterious portal that lets you visit the time and place of every nuclear device ever detonated, including the first Trinity test in New Mexico in 1945. Is there anything you can do to change the future?

Moonmist, Infocom's second entry, is modeled after gothic mystery novels. You're sent on a journey to a castle in England, where you become involved in a search for hidden treasure. Along the way you must deal with local superstitions and ghosts.

Commodore Amiga

After missing the Fall COMDEX and Winter CES—to the distress of its fans—Commodore made a big showing with the Amiga at the Spring COMDEX in Atlanta. However, a few weeks later, Commodore significantly scaled down its appearance at the Summer CES. Instead of going ahead with plans for a large exhibit on the main floor, Commodore switched to a small meeting room on an upper floor—the same meeting room occupied by Atari a year ago. Even more disappointing, the Amiga was nowhere to be seen. Commodore explained that it considers the Amiga to be a high-end personal/business computer, not a consumer computer, and therefore it came to CES with only the Commodore 128 and redesigned 64.

Nevertheless, several other companies introduced Amiga software at CES, and the big news at COMDEX was Commodore's announcement of a new IBM PC emulator—the Sidecar. The Sidecar is a plug-in expansion box, not to be confused with the currently available PC emulator, the *Transformer*. The *Transformer* emulates the PC entirely in software; the only hardware required is a 5¼-inch floppy disk drive. When the *Transformer* was finally released this spring after numerous delays, it became obvious that another solution would have to be found to make the Amiga truly IBM-compatible. The *Transformer* proved to be less compatible than its designers had hoped and was widely criticized for its slow execution speed.

As a result, Commodore decided to take the more conventional hardware approach to emulation. The Sidecar is basically an IBM PC without a keyboard. It's a large box that plugs into the expansion port, and it contains an 8088 microprocessor, an empty socket for an 8087 math coprocessor, 256K of RAM (expandable to 512K), a 5¼-inch disk drive, and three empty slots compatible with PC expansion

Lyco Computer Marketing & Consultants

ATARI

ATARI
XM 301 39

SUBLOGIC (Atari)
Flight Simulator 31 95
Night Mission Pinball 18 95
Scenery Disks EA 14 95

ATARI
1050 129
SF314 219
SF354 175

ATARI
130 XE Call
65 XE Call
520 st Call
520 st Monochrome Call
520 st Color Call
1027 Printer 145
1040 st (NEW) Call

SYNAPSE (Atari)
Synfile 29 95
Syncalc 29 95
Template 14 95

BRODERBUND (Atari)
Printshop 28 75
Graphics Lib I,II,III 18 75
Paper Refill 12 95
Karateka 19 75

UNISON WORLD
Printmaster 24 75
Art Gallery 18 75

FIREBIRD
The Pawn 26 75
Star glider 26 75

ACCESS
Leader board 24 75

VIP (520 St)
VIP Professional 109

SUPRA
Supra 300 39 95
Supra 1200 149 95

SSI (Atari)
Nam 24 75
Mechraged 34 95
Antietam 29 95
USAF 34 95

ACTIVISION (Atari)
Hackler 15 75
Mindshadow 15 75
Ghostbusters 15 75
Great Am. Race 15 75
Music Studio 20 75
Space Shuttle 15 75

ACTIVISION (520 St)
Borrowed Time 29 75
Music Studio 29 75
Hackler 26 75
Mindshadow 29 75

MICROLEAGUE (Atari)
Baseball 24 95
GM disk 24 95
Team disk 14 95

QUICKVIEW (520 St)
Zoomracks 49 95

HABA (520 St)
Writer 49 95

DISK NOTCHERS . . $7.95!!

apple

ACTIVISION (Apple)
Alter Ego 28 75
Little People 24 75
Mindshadow 24 75
Hackler 24 75
Gamemaker 24 75

BRODERBUND (Apple)
The Print Shop 31 50
Graphic Librarys EA 18 50
Bank St. Writer 128K 42 75
Bank St. Speller 42 75
Carmen Sandiego 22 75
Karateka 22 75
Captain Goodnight 22 75
Muppet Cruise 25 75
P S Companion 24 75
Science Kit 35 95

MICROPROSE (Apple)
Crusade in Europe 24 75
Decision in Desert 24 75
F-15 Strike Eagle 20 75
NATO Commander 20 75
Silent Service 20 75
Solo Flight 20 75

SSI (Apple)
Phantasie II 24 75
Wizard's Crown 24 75
Rings of Zilfin 24 75
Colonial Conquest 24 75
Battlegroup 35 75
NAM 29 75

MICROLEAGUE (Apple)
M L Baseball 24 95
General Mgr 24 95

COMMODORE

COMMODORE
128 CALL
C 1571 Drive CALL
C 1902-A CALL
C 1541 Drive CALL
C 1870 Modem CALL
C 1350 Mouse 39
C 1700 128 K RAM 145
C 1750 512 K RAM 269
JANE 35
Perfect Writer 45
Perfect Calc 45
Perfect Filer 45

COMMODORE
1571 CALL
1541 CALL

SYNAPSE
Syncalc 29 95
Template 14 95
Loderunner Rescue 19 95
Essex 24 95
Brimstone 24 95
Mindwheel 24 95

ACTIVISION (C-64/128)
Alter Ego 28 75
Hackler 18 75
Little People 20 75
Gamemaker 24 75
Borrowed Time 18 75
Space Shuttle 18 75
Music Studio ***24.75***
Mindshadow 18 75
Roadrace 18 75
Fast Tracks 22 75
Count Down 18 75

BRODERBUND
The Print Shop 24 75
Graphics Library I, II, III 15 75
Karateka 17 75
Bank St. Writer 29 75
Lode Runner 19 75
Printshop Companion 24 75
Bank St. Speller 29 75
Bank St. Filer 29 75
Bank St. Mailer 29 75
Championship Loderunner 19 75

SUBLOGIC
Nightmission Pinball 18 95
Flight Simulator 31 75
Jet Simulator 25 95
Football 25 95
Scenery Disk EA 14 95
Set 1-6 69 95

MICROLEAGUE
Baseball 24 95
GM Disk 24 95
Team Disk 14 95
Stat Disk 14 95

UNISON WORLD
Print Master (Amiga) 22 75
Print Master (C-128) 22 75
Print Master (C-64) 22 75
Art Gallery 16 75

BATTERIES INCLUDED
Paper Clip 35 95
Consultant 35 95
Paper clip w/ Spell Pak 48 95

FIREBIRD
Elite 19 95
Colossus IV 21 95

XETEC
Font Master II 64 28 95

CARDCO
Numeric Keypad 34 95
CB/5 5-slot Board 49 95
CB/2 2-slot Board 21 95
S'More Basic Rom 39 95
Write Now 29 95
Freeze Frame 29 95

MODEMS

SUPRA
1064 Modem (C-64) 49 95

COMMODORE
1670 Modem 155

ANCHOR
Volksmodem 55
Volksmodem 12 179

US ROBOTICS
Password 1200M 189
Password 1200F 229
Password 300M 139
Password 300F 139
Courier 2400 395

HAYES
Smartmodem 300 133
Smartmodem 1200 377
Smartmodem 1200B 347
Smartmodem 2400 598

SUPRA
1200 (520 ST) 149

DIGITAL DEVICES
Pocket Modem AT Call
Compuserve 18.95

COMPUSERVE . . 18.95

MONITORS

TEKNIKA
MJ-10 175
MJ-22 250
MS-305 RGB NEW

AMDEK
300 Green 118
300 Amber 128

COMMODORE
1902 Color CALL
1802 Color CALL

SAKATA
SG 1000 12" Green 99
SA 1000 12" Amber 109
SG 1500 12" Green TTL 119
SA 1500 12" Amber TTL 129
STS1 Tilt stand 29

THOMSON
CM36512V1 269
CM36632 159

PRINCETON GRAPHICS
MAX-12 Amber 175
HX-12 RGB 458
SR-12 RGB 575

ZENITH
ZVM 1220 95
ZVM 1230 95
ZVM 1240 149

NEW HOURS!
Mon-Thur 9AM-6PM
Fri 9AM-6PM
Sat 10AM-6PM

LYCO COMPUTER MARKETING & CONSULTANTS, INC.

NEW HOURS!
Mon-Thur 9AM-6PM
Fri 9AM-6PM
Sat 10AM-6PM

Lyco Computer Marketing & Consultants

NEW HOURS!
Mon-Thur 9AM-6PM
Fri 9AM-6PM
Sat 10AM-6PM

AMERICA'S MAIL ORDER HEADQUARTERS

NEW HOURS!
Mon-Thur 9AM-6PM
Fri 9AM-6PM
Sat 10AM-9PM

SAVE ON THESE IN STOCK PRINTERS

1080 .....$199

EPSON

LX80	209
FX85	369
DX10	207
HI80	355
HS80	298
FX286	529
LQ800	529
LQ1000	659

SEIKOSHA

SP-1000 VC (C-64)	185
SP-1000 A Centronics	197
SP-1000 I IBM	197
SP-1000 AS RS-232	197
SP-1000 AP Apple IIc	197
BP-5200 I	649
SP-1000 ribbon	8 50
BP-5200 ribbon	12 50
BP-5420	999

DIABLO

D25	549
D80 1 F	2395
P 32 CQ 1	699
P-38	1749
C-150	999
835	1029

NX-10....CALL

PANASONIC

1080	199
1091	228
1092	325
3131	257
3151	399
1592	439
1595	NEW

TOSHIBA

P1340	469
P351+	1149
P341P	969
P341S	999
354 Sheet Feeder	529

BROTHER

HR-15XL-P	359
HR-15XL-S	359

C. ITOH

1550 SP+	Call
D1040	Call
Prowriter Jr.	Call
Prowriter 8510 SP+	Call

STAR MICRONICS

NX-10 (NEW)	CALL
NB-15 (NEW)	CALL
SB-15 (NEW)	CALL
SG-15	367
SD-10	319
SD-15	438
SR-10	469
SR-15	578
SB-10	589
Powertype	297

CITIZEN

MSP-10	268
MSP-15	377
MSP-20	355
MSP-25	528
120-D	195
Premier 35	469

JUKI

Juki 6100	339
RS232 Serial Board	55
6100 Tractor	119
6100 Sheet Feeder	209
Juki 6300	757

OKIDATA

Okimate 10	179
182	214
192	348
193	563

LEGEND

1080	Call
1380	258
1385	289
808	148

SILVER REED

EXP 420 P	209
EXP 600 P	489
EXP 800 P	649

COLOR RIBBONS NOW AVAILABLE!!

DUST COVERS

Atari

520ST	11.95
130XE	6.99
800XL	6.99
1050	6.99
1025	7.99

Commodore

C128	7.99
1571/1541	6.99
1902	10.95
1702	8.99
C64/Vic20	6.99

Panasonic

1090/1091	8.99
1092	8.99
1093	9.99

DRIVES

INDUS
GT Commodore 185

TYMAC
MDD-640 3½"
Apple Drive 640K 289

COMTEL
Enhancer 2000 (C-64) 155

TANDON
320K ½ ¼" Drive 115

LOWEST PRICES

INTERFACING

CARDCO

G-WIZ (C-64)	47 95
Super G (C-64)	54
C/?PS (C-64)	37

MICROBITS

MPP-1150 (Atari)	54
MPP-1150XL (Atari)	59
MicroPrint (Atari)	39

TYMAC

Connection (C-64)	55
Tackler (Apple)	49
PPC-100 (Apple)	39

MICROTEK

Dumpling GX (Apple)	59
Dumpling 16K (Apple)	89
RV-611C (Apple)	49

ORANGE MICRO

GRAPPLER + (Apple)	85
Grappler 16K (Apple)	149
ORANGE (Apple)	59
Grappler CD (C-64)	79

XETEC

Super Graphix 64	64
Super Graphix JR 64	45

SUPER SAVINGS

DISKETTES

3.5" DISKETTES

3M

SSDD	19 95
DSDD	25 95

MAXELL

SSDD	18 99
DSDD	28 99

VERBATIM

SSDD	17 99
DSDD	25 99

5¼" DISKETTES

MAXELL

5¼" SSDD	11 99
5¼" DSDD	15 99

SUNKYOUNG

SKC 5¼" SSDD	11 99
SKC 5¼" DSDD	13 99

VERBATIM

5¼" SSDD	9 99
5¼" DSDD	13 99

BONUS

5¼" SSDD	6 99
5¼" DSDD	8 99

FASTEST SERVICE

IBM-PC

SUBLOGIC (IBM)

Jet Simulator	34 95
Scenery Disks EA	14 95
Set 1-6	69 95

BRODERBUND (IBM)

Bank St Writer	48 95
The Print Shop	34 95
Graphics Library 1	22 95
Ancient Art of War	27 95
Champ Lode Runner	22 95
Karateka	22 95

SYNAPSE (IBM)

Synstock	64 95
Essex	28 95
Wizard of Wall St	28 95
Brimstone	28 95

ACTIVISION (IBM)

Borrowed Time	24 75
Mindshadows	24 75
Music Studio	29 95
Alter Ego	29 95

MICROLEAGUE (IBM)

M L Baseball	24 95
General Mgr	24 95
85 Team Disk	14 95

LEADING EDGE

Nutshell	69 95
Nutshell Filer	149 00

TOLL FREE 1-800-233-8760

TO ORDER

CALL TOLL FREE 1-800-233-8760
In PA 717-494-1030
Customer Service 717-494-1670

or send order to
Lyco Computer
P.O. Box 5088
Jersey Shore, PA
17740

RISK FREE POLICY

In-stock items shipped within 24 hours of order No deposit on C O D orders Free shipping on prepaid cash orders within the continental U S Volume discounts available PA residents add sales tax APO, FPO, and international orders add $5 00 plus 3% for priority mail service Advertised prices show 4% discount for cash, add 4% for MasterCard and Visa Personal checks require 4 weeks' clearance before shipping Ask about UPS Blue and Red label shipping All merchandise carried under manufacturer's warranty Free catalog with order All items subject to change without notice

boards. A second floppy drive or 20-megabyte hard disk is optional, and there's also provision for up to two megabytes of Amiga memory expansion.

When the Sidecar is booted, two new icons labeled *PC Mono* and *PC Color* appear on the Amiga's Workbench screen. The Sidecar is designed to emulate the PC's monochrome and color/graphics modes, and clicking on one of these icons selects which mode to use. PC-DOS then opens up as a window on the Amiga Workbench screen. To the Amiga's multitasking operating system, the PC emulator is simply another task—so you can simultaneously run one or more Amiga programs while using the emulator. You can even open more than one PC window at once, if enough memory is available. You can't, however, multitask PC programs, since PC-DOS isn't a multitasking operating system.

Commodore says that this marriage of the PC and Amiga creates some interesting possibilities. For instance, you can plug a hard-disk expansion card into one of the Sidecar's slots and partition the disk for use with AmigaDOS as well as with PC-DOS. Amiga and PC software can run concurrently and exchange data using a common memory area. And although PC graphics are limited to four simultaneous screen colors as on a real IBM, you can select those four colors from the Amiga's much larger palette of 4096 colors.

The technology for the Sidecar originates from the two IBM PC clones which Commodore sells in Europe—the PC-10 and PC-20. (Commodore was going to introduce these machines into the U.S. market at Summer CES, but canceled its plans at the last minute.) Unlike the *Transformer*, the Sidecar is supposed to be nearly 100 percent IBM-compatible and capable of running programs at the full speed of a regular PC. At COMDEX, we saw the Sidecar running Microsoft's *Flight Simulator*, one of the toughest tests for any PC clone.

Scheduled for release this fall, the Sidecar is going to be priced relatively low. Although Commodore has not officially announced a price yet, indications are that it will cost $300 to $500.

Another interesting Amiga peripheral shown at COMDEX was the FutureSound digital sound recorder from Applied Visions (Medford, Massachusetts). The package comes with a digitizer, microphone, recording software, and a cable that plugs into the parallel printer port. A phono jack on the digitizer allows you to bypass the microphone for direct recording or to mix two different sound sources. Any sound can be recorded and played back at any speed, and recorded sounds can also be played by your own programs written in C or Amiga BASIC. The sampling rate can be varied from a few samples per second to 28,000 samples per second (the higher the rate, the greater the quality—and the more memory required). Price: $175.

An Amiga expansion box was announced by The Gemstone Group (Buffalo Grove, Illinois). Current plans call for eight expansion slots, 512K of RAM (expandable to eight megabytes), a hard disk interface, and a realtime clock with battery backup. The Gemstone Group also is considering a CD-ROM interface and MIDI ports as additional standard features. The box is scheduled for release late this summer for $995. A version with eight megabytes of RAM installed is tentatively priced at $1,995.

Golden Hawk Technology (Nashua, New Hampshire) announced a MIDI interface with in/out jacks and a synchronization connector for controlling drum machines and other devices. It hooks up to the serial port and is priced at $79.95.

Amiga musicians will also be interested in *SoundScape Pro*, a MIDI sequencer system from Mimetics Corporation (Palo Alto, California). *SoundScape Pro* uses the Amiga's multitasking operating system to make multiple music programs behave like separate pieces of studio equipment, all tied together through a software patch panel. It provides the equivalent of a MIDI clock generator, a sampling synthesizer, and a digital tape deck. The price is $149. Mimetics also is releasing the SoundScape Digital Sampler for $99 and a MIDI interface for $49.

Flow, an idea processor from New Horizons Software (Austin, Texas), is designed to help you create and organize presentations, reports, projects, and events. It takes advantage of the Intuition user interface, but also provides keyboard shortcuts. Price: $99.95.

Byte by Byte (Austin, Texas) announced two Amiga programs: *InfoMinder*, a hierarchical database manager, and *Write Hand*, a word processor. *InfoMinder* is unique in that it lets you combine text and graphics, and it also can be used to program custom applications. Price: $89.95. *Write Hand* has online help screens and is designed to make it easy for small businesses to generate form letters. Price: $50.

Electronic Arts announced several programs to be available this summer, including *Chessmaster 2000* ($44.95); *DeluxePaint Art & Utility Disk #1*, a supplement to the popular *DeluxePaint* ($29.95); *DeluxePrint Art Disk #2*, a supplement to *DeluxePrint* ($29.95); *DeluxeVideo*, the long-awaited presentation graphics program ($99.95); *Instant Music*, a composition tool for nonmusicians ($49.95); *Marble Madness*, an arcade-style game ($49.95); and *Ultima III*, an adventure game ($59.95).

Access Software (Woods Cross, Utah) is introducing its hit golf simulator, *Leader Board*, for the Amiga. As realistic as this program is on the Commodore 64—with 3-D animation, true perspective view, detailed landscapes, and lifelike sounds—it should be even better on the Amiga. The price is $39.95.

Master Designer Software, in cooperation with Mindscape, (Northbrook, Illinois), is bringing out a series of five new games for the Amiga in late 1986/early 1987 under the Cinemaware label. These games are described as interactive movies that combine classic movie themes with sophisticated computer graphics. All are role-playing games, and the graphics imitate film effects such as 3-D movement, zooms, cuts, pans, close-ups, and changes in perspective. The titles scheduled so far include *Sinbad and the Throne of the Falcon*, *The King of Chicago*, *S.D.I.*, *Defender of the Crown*, and *Star Rush*. They'll also be available on the Atari ST and Apple Macintosh. ©

SILICON EXPRESS DISKS SS/DD 7.95 DS/DD 8.95

VERBATIM OR MAXELL SS/DD 13.95 DS/DD 16.95

ELEPHANT SS/SD 7.95 SS/DD 9.95 DS/DD 9.95

DATACASE 8.95

INSIDE OHIO 1-800-225-0755

OUTSIDE OHIO

YOUR TOLL FREE ACCESS

1-800-228-0755

CALL...FOR THE BEST PRICES...
BEST DELIVERY...
We have hundreds of other programs in stock at Fantastic savings.

SILICON EXPRESS

ALL NEW STORE WITH EXPANDED WAREHOUSE

	IBM
Ability	299.95
ASC II Express	132.95
Bank Street Writer	52.95
Clout	119.95
Copy II Option Bd	74.00
Cornerstone	259.95
Crossstalk	94.95
dBase III+	389.95
Deadline Manager	104.95
Dollars & Sense	104.95
Easy	97.50
Einstein Writer	119.95
Electric Disk	239.95
Fontrix 1.5	84.95
Fntpks 1 thru 13 ea	13.00
Framework II	389.95
Gamecard III	39.95
Gato	23.99
In House Accountant	119.95
Jet	32.95
Knowledgeman 2.0	319.95
Lotus	CALL
Mngng Yr Money	104.95
Micro Cookbook	25.95
Microsoft Chart 2.0	299.95
Word	239.95
Windows	99.95
Morton Utilities	55.95
PFS: Access	59.95
PFS: Files	92.50
PFS: First Success	229.95
PFS: Graph	92.50
PFS: Plan	92.50
PFS: Proof	59.95
PFS: Report	92.50
PFS: Write	92.50
Print Shop	36.95
Print Shop Gr. #1	22.95
Print Shop Refill	13.95
R:Base 5000	339.95
Reflex	56.95
Remote	94.95
Report Card	33.00
Samma +	369.95
Sidekick protected	29.95
Sidekick not protected	47.95
Sideways Ver. #3	49.95
Speedkey	104.95
SuperCalc 3	239.95
Superkey	35.50
Sprky/Sidek Combo	97.95
Symphony	CALL
Think Tank	99.95
Time Is Money	79.99
Traveling Sidekick	36.95
Turbo Database Tlbx	39.95
Turbo Lightning	59.95
Turbo Pascal 3.0	39.50
Turbo Pascal 8087	59.95
Turbo Prolog	69.95
Turbo Tutor	24.95
Ultima III	34.95
Visable 8088	49.95
Volkswriter Dlx #3	159.95
Web Spell Checker	41.95
Will Writer	31.95
Wizardry	39.95
Word Perfect	219.95
Wordstar	214.95

APPLE SOFTWARE	
Adv. Construction	30.75
Algeblaster	34.95
Alpha Plot	24.99
ASC II Express	72.50
Auto Works	27.95
Bag of Tricks 2	29.00
Bank Street Filer	39.95
Bank Street Mailer	39.95
Bank Street Writer	39.95
Bank Street Speller	39.95
Bard's Tale	29.50
Beagle Bag	14.95
Beagle Basic	17.95
Beagle Graphics	29.95
Beneath Apple DOS	13.95
Big U	18.95
Bookends	64.95
Captain Goodnight	19.95
Carmen Sandiego	22.45
Chmpshp Ld. Runner	19.45
Chart'n Graph Tlbx	23.95
Computer Quarterback	27.95
Crossword Magic	30.00
Crypt of Medea	21.95
Database Toolbox	23.95
Dazzle Draw	33.95
D-Code	19.95
DiskQuick	14.95
Dollars & Sense	86.60
DOS Boss	11.95
Double-Taka	17.95
Eidolon	23.50
Essential Data Dup	41.95
Extra K	19.95
Fantavision	28.95
Fat Cat	17.95
Flex Type	14.95
Fontrix 1.5	53.25
Fntpks 1 thru 13 ea	13.00
Font Works	34.95
Forecast	38.00
Frame-Up	14.95
Gato	20.45
G I Joe	19.95
Graphics Department	84.95
Graphics Expander	21.95
Graphworks	57.95
GPLE	24.95
Handler Package	89.95
Impossible Mission	23.50
I.O. Silver	14.95
Jeeves	29.95
Karateka	19.45
Knight of Diamonds	19.50
Kids on Keys	17.95
Kid Writer	18.99
King's Quest I or II	29.95
Koronis Rift	23.49
Legacy of Llylgamyn	22.50
List Handler	39.95
Locksmith	59.95
Mngng Yr Money	102.00
Mathmaze	27.95
Megaworks	64.95
Merlin	33.89
Merlin Pro	54.95
Merlin Combo	83.95
Micro Cookbook	21.45
Millionaire	22.50
Minipix	14.99
Moebius	34.95
MouseWrite	71.95

APPLE SOFTWARE	
Multiplan DOS 3.3	95.95
Munch-A-Bug	23.95
NATO Commander	19.45
PFS: Plan	89.95
PFS: Access	44.00
PFS: Write	69.95
PFS: File	69.95
PFS: Report	69.95
PFS: Graph	69.95
PFS: First Success	129.00
Piece of Cake Math	15.95
Pinpoint	39.95
Prince	49.95
Printographer	23.95
Print Shop	29.99
Prt Shp Gr #1,2, or 3	13.95
Print Shop Comp	21.95
Probyter	17.95
Pronto DOS	14.95
Prot. tour golf	25.99
Quicken	51.95
Report Card	32.95
Rescue/Fractalus	23.50
Reportworks	84.95
Rescue Raiders	24.95
Science tool kit	32.95
Sensible Grammar	52.95
Sensible Speller	84.95
Shape Mechanic	23.95
Sideways	29.95
Silicon Salad	12.69
Skyfox	24.50
Spellworks	29.95
Spitfire Simulator	24.75
Sticky Bear ABC	23.75
Basket Bounce	23.75
Car Builder	23.75
Math	23.75
Numbers	23.75
Opposites	23.75
Reading	23.75
Spellgrabber	23.75
Townbuilder	23.75
Typing	23.75
Summer Games I or II	22.45
SuperCalc 3A	109.95
Terrapin Logo	59.95
Think Tank	98.89
Thinkworks	64.95
Time is Money	57.45
Tip Disk #1	12.00
Transylvania	19.50
Triple-Dump	16.95
Turbo Database Tlbx	27.95
Turbo Pascal 3.0	34.95
Turbo Tutor	20.50
Typefaces	10.89
Ultima III	33.25
Ultima IV	33.25
Understanding IIE	17.95
Universal file conv	24.95
Utility City	15.25
Versaform	43.45
Video Toolbox	23.95
Visable 8502	31.75
Wilderness	39.95
Winter Games	21.45
Wizardry	28.50
Wizard's Toolbox	23.95
Wizprint	14.25
Word Handler	39.95
Word Perfect	125.95
World's Grtst Fball	22.45
Write Choice	28.45

COMMODORE	
Acrojet	24.95
Air Rescue I	24.95
Bard's Tale	28.50
Ballblazer	23.49
Back to Basics	134.95
Bank Street Storybook	27.95
Bank Street Speller	34.95
Bank Street Writer	34.95
Basic Toolkit	29.95
Blazing Paddles	24.95
Brrwed Time 64/128	20.95
Color Me	20.95
Cntdwn to Shtdwn	29.95
Crossword Magic	30.00
Fast Tracks	20.95
Gertrude's Secrets	20.95
G I Joe	19.95
Graphics Magician	27.95
Grappler CD	94.95
Hacker	20.95
Jet 64/128	27.95
Jet Combat Simulator	20.95
Karateka	19.45
Kennedy Approach	24.95
Kola Pad	54.95
Kung Fu	20.95
Little Comp. People	24.95
Mail Order Monsters	24.95
Merlin	34.95
Mig Alley Ace	24.95
Multiplan 64/128	43.95
Music Shop	31.95
Nato Commander	19.45
Paper Clip	62.95
PFS: File	54.95
Print Shop	29.95
Print Shop Comp	21.95
Pr. Shop Gr. 1,2,or 3	13.95
Rescue on Fractalus	23.50
Road Race	20.95
Scrabble 64/128	27.95
Sideways	20.95
Silent Service	19.45
Spitfire Ace	20.95
Spreadsheet	34.95
Spy vs. Spy	20.95
Sticky Bear ABC	23.75
Numbers	23.75
Opposites	23.75
Shapes	23.75
Summer Games I or II	22.45
Temple of Apshai Trl.	22.45
Winter Games	21.45
World's Grtst Bsball	19.45
World's Grtst Fball	22.45

IBM / APPLE / COMMODORE	
Clip Art (Vol 1)	15.95
Clip Art (Vol 2)	21.95
Copy II+	19.50
Crusade in Europe	22.95
Cut & Paste	23.00
Cutthroats	23.89
Deadline	29.00
Early Games	15.95
Easy as ABC's	20.95
Enchanter	23.69
Family Roote	149.95
Flight Sim II	33.99
Fraction Factory	15.95
F-15 Strike Eagle	19.45
Ghost Busters	22.00
Hitchhiker's Guide	22.00
Infidel	26.00
Load Runner	19.45
Magic Spells	20.75
Mastertype	24.99
Math Blaster	28.95
Micro Lg. Baseball	22.50
Micro Lg. Manager	27.95
Micro Lg. Team	13.95
Mind Forever Voy	30.95
Music Construction	24.75
Newsroom	30.50
One-On-One	24.50
Pitstop II	23.50
Planetfall	22.00
Print Shop Refill	12.25
Reader Rabbit	25.95
Sargon III	27.20
Sat (Harcort Brace)	46.99
Seastalker	22.00
Sorcerer	26.00
Solo Flight	19.45
Speed Reader II	33.95
Spell It	26.95
Spy Hunter	29.00
Starcross	29.00
Suspect	28.00
Suspended	28.00
Temple of Apshai Trl	22.45
Typing Tutor III	29.75
Will Writer	28.45
Wishbringer	27.95
Witness	22.00
Word Attack	26.95
Word Spinner	23.95
World's Grtst Bsball	19.45
Zork I	22.00
Zork II or III	25.00

HARDWARE	
Compuserve Starter	21.95
Datacase	8.99
Disk Notcher	4.95
Flip-n-File	14.95
Lemon	31.95
Lime	49.95

APPLE HARDWARE	
80 Col. 64K Card	49.95
Apple Cat II	194.95
Gibson Light Pen	149.00
Grappler Buffered	130.95
Grappler +	71.95
Grappler (serial)	71.95
Grappler C	71.95
Hotlink	54.95
Kraft Joystick	25.95
Koala Pad +	70.45
Mach II Joystick	29.20
Mach III Joystick	32.20
Micromodem IIE	139.95
Microsci A2 Drive	144.95
Microsci IIC Drive	149.95
Mockingboard A or B	65.95
Mockingboard C	118.45
Mockingboard D	128.45
Mockingboard A-MAX	39.95
Multiram CX 512K	209.95
Multiram IIE 90/64K	124.95
Multiview II+ 90/160	199.95
Paddlesticks	26.50
Parallel Printer Card	54.95
Prometheus 1200A	254.95
Print-it	128.00
Super Serial Card	90.95
System Saver Fan	59.95
Thunder Clock	102.95
Wildcard 2	74.95
ZEE 90A Checkmate	93.95
Zoom IIE	109.00

PRINTERS	
Star NX-10	259.00
Star SG-15	394.99
Star Powertype	344.95
Printer Stand	16.95
Okidata 192P	229.95
Okidata 193P	399.95
Okimate 20	209.95
Colored printr ribbons	CALL
Color Paper Pack	13.95

MONITORS	
Magnavox Monitors	CALL
Amdek Color 600	414.99
Amdek Video 300G	127.99
Amdek Video 300A	137.99

YOUR ORDER FORM
SILICON EXPRESS
5955 E. Main Street Columbus, Ohio 43213

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

CHARGE CARD # _____ Exp Date _____

QTY	DESCRIPTION	PRICE

Computer Type _____ SHIPPING

Phone No _____ TOTAL

No additional surcharge on Mastercard, Visa, or School purchase orders. Personal checks allow 2 weeks. Shipping charges are $3.00 minimum. Ohio residents add 5½% state sales tax. Prices subject to change without notice.

SILICON EXPRESS

5955 E. Main St. Columbus, Ohio 43213

1-614-868-6868

TIGHTROPE

Daniel Aven

Stretched high above the circus arena, the tightrope beckons. Can you walk all the way across the rope without falling into the net? This interesting program is both an arcade game and a typing tutor. The original version was written for the Commodore 64 and also works on the Commodore 128 in 64 mode. We've added new versions for the Apple, IBM PC/PCjr, Amiga, and Atari 400, 800, XL, and XE computers. The Commodore 64 version requires a joystick. The IBM PC/PCjr version requires a color/graphics card and BASICA for the PC, and Cartridge BASIC for the PCjr. The Amiga version requires 512K of memory and Amiga BASIC. The Atari version requires at least 32K of memory and a joystick. The Apple version works with either a joystick or game paddles.

Arms outstretched, you venture cautiously onto the tightrope. The rope quivers for a moment, then steadies. Far below, in a packed circus tent, the crowd roars its encouragement. Don't worry, there's a safety net below. But you won't entertain the onlookers—or earn points in this game—by falling into the net. Your first few steps are hesitant, but with practice your progress becomes more sure. After what seems an eternity, you reach the other side. After cheering its approval, the crowd cries out for a repeat performance.

"Tightrope" combines a novel game idea and realistic animation with an educational goal. You can play it either as an arcade game or as a typing tutor. In game mode, the object is to walk all the way across the tightrope without falling into the net. In tutorial mode, you must watch for a letter to appear next to the acrobat's head, and type it on the keyboard before time runs out.

Type in and save a copy of Tightrope, referring to the special instructions for your computer. When you run the program, it asks you to choose between a game of skill and a typing tutorial.

A Delicate Balance

If you choose the game of skill, your goal is simple: Move the animated acrobat all the way across the tightrope without falling into the net. As the acrobat walks along, you'll occasionally begin to fall to one side or the other. But there's always time to recover your balance by pushing the joystick (or keyboard controls in some versions) in the opposite direction of the fall. If you countermove just enough to regain your balance, all is well and the acrobat begins to walk again. If you move too far in the opposite direction, the teetering starts all over again.

It's a delicate balancing act, and it grows more difficult each time you make it across the rope. When you succeed in reaching the opposite side, you advance to the next skill level. At each higher level, it becomes more and more difficult to keep your balance.

If you lose your balance completely, the acrobat falls to the safety net and bounces a few times before coming to rest. At this point you can try again at the same level or return to the main menu to choose a different game.

Your score is based on how far you get before falling. Each successful step is worth a certain number of points, and this value increases at higher levels. In addition, bonus points are awarded for rapid progress; the faster you move across the rope, the higher the bonus.

Typing Tutor

In the tutorial version of Tightrope, the object is the same—move the acrobat across the rope without falling—but different means are used to keep your balance.

When you see a character appear next to the acrobat's head, that's your cue to press the corresponding key on the keyboard. If you type the correct letter, the acrobat straightens up. If you press the wrong key, a buzzer sounds and the acrobat teeters even more.

To remain on the rope, you must continue to type the same letters that appear on the screen. In other respects, the tutorial version of Tightrope is the same as the skill game.

NEW NEW NEW

$395.00 128K Apple $395.00 Compatible Computer

The moment you buy it the Laser 128 computer is compatible with virtually all software for the Apple® II, Apple®IIe, and Apple®IIc. All the equipment is here. Plus the Laser 128 gives you the features Apple® forgot.

• 5¼" Disk Drive • Additional Disk Drive Interface • Serial Printer Interface • Parallel Printer Interface • Modem Interface • RGB 80 Column Color Interface • Mouse Interface • Hi-Res Graphics • 32K Rom • 128K RAM • Expansion Slot for Apple® Peripheral cards • Battery Pack Interface • Numeric Keypad • Programable Function keys • Carrying Handle

Limited time only

*Free Magic Window IIe Professional Word processor ($149.95 Value)

*with Laser 128 Purchase

$395.00 $395.00

Comparably priced Apple®IIe system **$1500.00**

Comparably priced Commodore® 128 system **$850.00**

- Home
- Business
- Students
- 90 Day Warranty
- 15 Day Free Trial

This is the best computer buy in the U.S.A. You get the complete Laser 128 computer with oll the features shown, Plus for a limited time only, you get the Professionol Mogic Window IIe Wordprocessor all for only $395.00. We hove tested this machine with over 500 software titles from our comprehensive catalog and the Laser 128 runs them oll perfectly. (it even runs Appleworks®) This is a must buy for new and old computer buyers. List $648.95. **Sale $395.00.**

	List	Sale
Centronics cable (for Centronics Printer)	$ 29.95	$ 19.95
Serial Cable (for Modem hookup)	$ 34.95	$ 19.95
9" 80 Column Hi-Res Green Monitor	$199.00	$ 99.95

Accessories	List	Sale
2nd Disk Drive	$299.95	$129.95
Professional Analog Joystick	$ 39.95	$ 19.95
RGB cable (RGB Monitor hookup)	$ 24.95	$ 19.95

Apple and Commodore are registered trademarks of Apple Computer Inc. and Commodore Business Machines Inc. respectively

Add $10.00 for shipping, handling, and insurance. Illinois residents please add 6¼% sales tax. Add 20.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. All orders must be in U.S. Dollars. WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA. Enclose Cashier Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail. Prices & Availability subject to change without notice.

VISA — MASTER CARD — C.O.D. C.O.D. on phone orders only.

Computer Direct

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

COMMODORE 64 COMPUTER

(Order Now)

$139.95

- C128 Disks 79¢ ea.*
- Commodore Writer 64 $19.95
- 13" Color Monitor $139.95

COMMODORE 64 SYSTEM SALE

Commodore 64 Plus $30.00 S&H

Com. 1541 Disk Drive

13" Color Monitor

$457

C128 Commodore Computer & 1571 Disk Drive

$499.00

- Voice Synthesizer $39.95
- 12" Monitor $79.95

CALL BEFORE YOU ORDER

PRICES MAY BE LOWER

COMMODORE 64 COMPUTER $139.95

You pay only $139.95 when you order the powerful 84K COMMODORE 64 COMPUTER! LESS the value of the SPECIAL SOFTWARE DISCOUNT COUPON we pack with your computer that allows you to SAVE OVER $250 off software sale prices!! With only $100 of savings applied, your net computer cost is $39.95!!

*** C128 DOUBLE SIDED DISKS 79¢ EA.**

Get these 5¼" Double Sided Flappy Disks specially designed for the Commodore 128 Computer (1571 Disk Drive). 100% Certified, ***Lifetime Warranty,*** Automatic Lint Cleaning Liner included. 1 Box of 10 - $9.90 (99¢ ea.), 5 Boxes of 10 - $44.50 (89¢ ea.), 10 Boxes of 10 - $79.00 (79¢ ea.).

13" COLOR MONITOR $139.95

You pay only $139.95 when you order this 13" COLOR MONITOR. LESS the value of the SPECIAL SOFTWARE DISCOUNT COUPON we pack with your monitor that allows you to save over $250 off software sale prices!! With only $100 of savings applied, your net color monitor cost is only $39.95. (16 Colors).

Premium Quality 150-170 CPS Comstar Aero 160 Printer $199.00

The COMSTAR Aero 160 gives you a 10" carriage, 150-170 CPS, 9 x 9 dot matrix with double strike capability for 18 x 18 dot matrix (near letter quality), high resolution bit image (120 x 144 dot matrix), underlining, back spacing, left and right margin setting, true lower decenders with super and subscripts, prints standard, block graphics and special characters. It gives you print quality and features found on printers costing twice as much!! (Centronics Parallel Interface) List $499.00 **Sale $199.00.**

9" SAMSUNG GREEN SCREEN MONITOR

Super High Resolution composite green screen monitor. Perfect for 80 column use with The C128 computer (Req. $19.95 Cable)
List $129.95 **Sale $59.95.**

80 COLUMNS IN COLOR
COMMODORE WRITER 64 WORD PROCESSOR $19.95

This COMMODORE WRITER 64 WORD PROCESSOR is the finest available for the COMMODORE 64 computer! The ULTIMATE FOR PROFESSIONAL Word Processing, DISPLAYS 40 or 80 COLUMNS IN COLOR or black and white! Simple to operate, powerful text editing, complete cursor and insert/delete key controls line and paragraph insertion, automatic deletion, centering, margin settings and output to all printers! List $99.00, **SALE $19.95.** ***Ltd. Qty Closeout Item***

SPECIAL SOFTWARE COUPON

We pack a SPECIAL SOFTWARE DISCOUNT COUPON with every COMMODORE 64 COMPUTER, DISK DRIVE, PRINTER, or MONITOR we sell! This coupon allows you to SAVE OVER $250 OFF SALE PRICES!!

(Examples)

PROFESSIONAL SOFTWARE
COMMODORE 64

Name	List	Sale	Coupon
PaperClip	$59.95	$34.95	$39.95
Consultant	$59.95	$49.95	$39.95
Leader Board	$39.95	$24.95	$22.95
The Print Shop	$44.95	$27.95	$26.95
Hailey's Project	$39.95	$22.95	$19.95
Practicalc (spread sheet)	$59.95	$19.95	$14.95
Voice Command Module	$79.95	$39.95	$34.95
Nine Princes in Amber	$32.95	$24.95	$21.95
Super Bowl Sunday	$35.00	$22.95	$19.95
Flip and File Disk Filer	$24.95	$14.95	$12.95
Pro Jay Stick	$19.95	$12.95	$10.00
PartyWare	$19.95	$14.95	$11.95
Dust Cover	$ 8.95	$ 6.95	$ 4.60
Financial Planner Sylvia Porter	$59.95	$38.95	$35.95
Hardball	$29.95	$18.95	$16.95
C64 Troubleshoot & Repair Guide	$24.95	$15.95	$12.95

(See over 100 coupon items in our catalog)

Write or call for Sample SPECIAL SOFTWARE COUPON!

ATTENTION
Computer Clubs
We Offer Big Volume Discounts
CALL TODAY!

PROTECTO WARRANTY

All Protecto's products carry a minimum 90 day warranty. If anything fails within 90 days from the date of purchase, simply send your product to us via United Parcel Service prepaid. We will IMMEDIATELY send you a replacement at no charge via United Parcel Service prepaid. This warranty proves once again that ***We Love Our Customers.***

C128 COMPUTER & 1571 $499.00

Now you can get the C128 Commodore computer & the 1571 Disk Drive for one low price of only $499.00. List $698.00
SALE $499.00.

340K 1571 COMMODORE DISK DRIVE $259.00

Double Sided, Single Disk Drive for C-128 allows you to use C-128 mode plus CPM mode. 17 times faster than 1541, plus runs all 1541 formats.
List $349.00. **Sale $259.00**

SUPER AUTO DIAL MODEM $29.95

Easy to use. Just plug into your Commodore 64 computer and you're ready to transmit and receive messages. Easier to use than dialing your telephone, just push one key on your computer! Includes exclusive easy to use program for up and down loading to printer and disk drives.
Best in U.S.A. List $99.00. **SALE $29.95.**
Coupon *$24.95.*

VOICE SYNTHESIZER $39.95

For Commodore-64 computers. Just plug it in and you can program words and sentences, adjust volume and pitch, make talking adventure games, sound action games and customized talkies!! PLUS ($19.95 value) TEXT TO SPEECH program included FREE, just type a word and hear your computer talk — ADD SOUND TO "ZORK". SCOTT ADAMS AND OTHER ADVENTURE GAMES!! (Disk or tape.) List $89.00. **SALE $39.95**

12" MAGNAVOX (NAP) 80 COLUMN MONITOR WITH SOUND $79.95

Super High Resolution green screen monitor. 80 columns x 24 lines, easy to read, plus speaker for audio sound included. Fantastic value.
List $129.00. **Sale $79.95.**
(C128 cable $19.95. C64, Atari cable $9.95)

PRINTER/TYPEWRITER COMBINATION $229.95

Superb letter quality, daisy wheel printer/typewriter combination. Two machines in one — just a flick of the switch. Extra large carriage, typewriter keyboard, automatic margin control, compact, lightweight, drop in cassette ribbon! (90 day warranty) centronics parallel interface. Built-In. List $349.00. **SALE $229.95.** ***(Ltd. Qty.)***

14" RGB & COMPOSITE COLOR MONITOR $259.95

Must be used to get 80 columns in color with 80 column computers (C128 - ISM - Apple). (RGB Cable $19.95) Add $14.50 shipping
List $399.00. **SALE $259.95.**

- LOWEST PRICES • 15 DAY FREE TRIAL
- BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL

PHONE ORDERS
8 a.m. - 8 p.m. C.S.T. Weekdays
9 a.m. - 12 noon C.S.T. Saturdays

- 90 DAY FREE REPLACEMENT WARRANTY
- OVER 500 PROGRAMS • FREE CATALOGS

Add $10.00 for shipping, handling, and insurance. Illinois residents please add 6¼% sales tax. Add 20.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. All orders must be in U.S. Dollars. WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA. Enclose Cashier Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail. Prices & Availability subject to change without notice.

VISA — MASTER CARD — C.O.D. C.O.D. on phone orders only.

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

Famous Comstar National Brand

10" Printer Sale

Includes Commodore Interface

Near Letter Quality *Near Letter Quality*

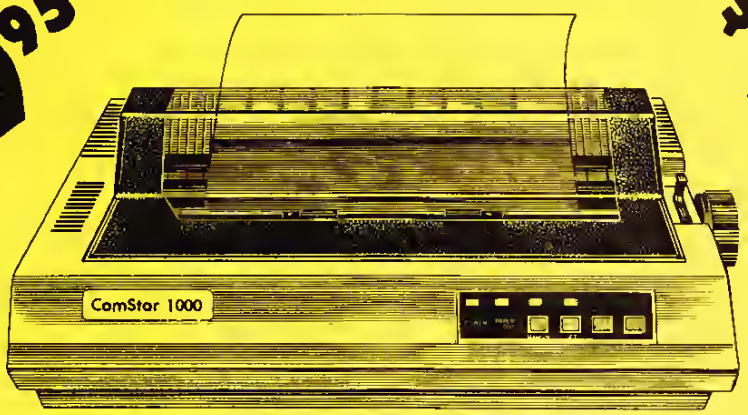
Best Value in the USA

• 100 CPS draft/20CPS near-letter quality • Dot Addressable Graphics • Adjustable Tractor and Friction Feed • Automatic Paper Loading • Right and Left Margin settings • Pica, Elite, Condensed, Italics • Superscript • Subscript • Underline, Bold print, Double Strike • Superb NEAR LETTER QUALITY

$179.95

Easy to Use

Fantastic Graphics



$179.95

2 Year *limited* **Warranty**

The Comstar 1000 is one of the best values in the United States today. Print your letters, documents, programs, pictures, and more at a blazing 100 Characters Per Second or 20 cps in the Near Letter quality mode. (Looks just like it came from a typewriter.) Plus, choose your printing mode (NLQ, Draft) from your wordprocessor or by simply pushing the controls on the front panel. Fantastic Quality at a Fantastic Price.List $349.00 SALE$179.95.

Print Method
Serial Impact dot matrix (9 pin)
Print Speed
Draft- 100 CPS NLQ- 20 CPS
Character Sets
96 ASCII Characters, Marker, Symbols (Includes Italic font)

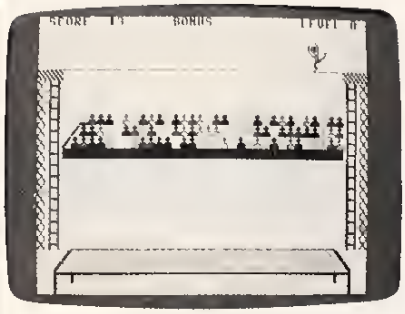
Ribbon (Life exp.)
Black: cassette (2.5 million characters)
Dimensions
15.4 (W) x 10.9 (D) x 4.7 (H) inch
Weight
Approx. 10 lbs
Character Spacing
Fixed

Line Spacing
1/6, 1/8, 7/72, and 1/216 inch
Paper Feed
Adjustable tractor and friction feed
Paper feeding Direction
Bi-directional
Copies
2 plus original

Supply is Limited so Buy Today

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery. 2 to 7 days for phone orders. 1 day express mail!
VISA — MASTER CARD — C.O.D.
No C.O.D. to Canada, APO, FPO

PROTECTO
We Love Our Customers
22292 N. Pepper Rd., Barrington, Illinois 60010
312/382-5244 to order



"Tightrope" for the Commodore 64, an amusing and educational game.

Commodore 64/128 Version

This version of Tightrope requires a joystick plugged into port 2. After you finish walking across the rope, you must repeat the performance while riding a unicycle. At successive skill levels, the acrobat alternates between walking and riding the unicycle.

Atari Version

Tightrope for the Atari requires a joystick plugged into port 1 and at least 32K of Random Access Memory (RAM). Move the joystick right or left to balance the acrobat.

IBM PC/PCjr Version

This version of Tightrope requires a color/graphics card and BASICA for the PC, or Cartridge BASIC for the PCjr. Play the game with keyboard controls: Press the Z key to move left (your left, not the acrobat's), and the slash (/) key to move right.

Amiga Version

Tightrope for the Amiga requires at least 512K of RAM. When typing the program listed below, do not type in the left-arrow symbol at the end of each line; it's there only to show you where the line ends (we deliberately chose a character that's not available from the Amiga's keyboard). Instead, wherever you see a left-arrow in the Amiga listing, press RETURN.

The Amiga game uses the same keyboard controls as the IBM PC/PCjr version: Press Z to move left and the slash (/) key to move right.

Apple II Version

The Apple version of Tightrope works with either a joystick or game paddles and runs on any Apple II-series computer with either ProDOS or DOS 3.3.

For instructions on entering these listings, please refer to "COMPUTE!'s Guide to Typing in Programs" in this issue of COMPUTE!.

Program 1: Commodore 64/128 Tightrope

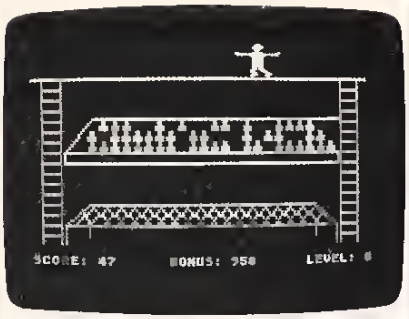
```
RM 10 U1=54296:U2=54277:U3=542
      78:U4=54276:U5=54273:U6=
      54272
CQ 20 ML$="[I]"+CHR$(8)+"[X]<"
      +CHR$(3)+"[2]XJ"+CHR$(16
      )+CHR$(248)+"L[B][T]":PO
      KE835,0
MJ 30 POKE836,208:POKE830,0:PO
      KE831,216:POKE828,0:POKE
      829,64:POKE56334,0
GK 40 POKE1,51:ML$=ML$:SYS(PEE
      K(51)+256*PEEK(52)):POKE
      1,55:POKE56334,1
MJ 50 POKE53272,30:FORA=14336T
      O14343:REAOB:POKEA,B:NEX
      T:FORA=0TO6:REAOC(A):NEX
      T
PH 60 GOTO80:OATA24,60,60,24,1
      26,255,255,255,7,0,1,3,5
      ,2,4
MR 70 FORA=1TOX:POKE646,C(RNO(
      1)*7):PRINT"@";:NEXT:PRI
      NT:RETURN
QE 80 PRINT"{CLR}{PUR}";:POKE5
      3280,4:POKE53281,1
FB 90 PRINT"{CLR}{9 DOWN}"SPC(
      15)"1- GAME"
KS 100 PRINT"{2 DOWN}"SPC(15)"
       2- TYPING"
AP 110 GETA$:IFA$<>"1"ANOA$<>"
       2"THEN110
SM 120 W=VAL(A$)
PA 130 OATA0,0,0,0,0,0,0,56,0,
       0,92,0,0,252,0,0,92,0,1
       28,56,8,127,255,240,0,1
       68
PH 140 OATA0,0,216,0,0,248,0,0
       ,136,0,0,248,0,0,248,0,
       0,248,0,0,240,0,1,192,0
XG 150 DATA0,224,0,0,176,0,0,1
       52,0,1,176,0
SB 160 OATA1,240,0,3,176,0,1,2
       0B,0,0,112
RS 170 OATA0,0,48,0,0,48,0,0,9
       6,0
HS 180 OATA1,240,0,1,240,0,3,1
       76,0,3,48,0,1,176,0,0,1
       52,0,1,176,0
FG 190 PRINT"{3 DOWN}
       {5 SPACES}ENTER LEVEL O
       F OIFFICULTY 0-9
CS 200 GETA$:IFA$=""THEN200
OH 210 IFA$<"0"ORA$>"9"THEN200
QX 220 B=VAL(A$):O=B:B=10-B
SF 230 P1=206:Y2=69:PX=201:PZ=
       205
RA 240 IFW=2THENB=B*8
PB 250 PRINT"{CLR}{REO}
       {5 DOWN}[3 +]{BLK}
       [34 T]{RED}[3 +]";
ER 260 FORX=1TO15:PRINT"{GRN}V
       {BLU}[Q][W]"SPC(34)"
       {BLU}[Q][W]{GRN}V";:NEX
       TX
GC 270 PRINT"{DOWN}{2 UP}
       {4 RIGHT}{BLK}[32 @]"
RH 280 PRINT"{3 RIGHT}N{YEL}VV
       VVVVVVVVVVVVVVVVVVVVVV
       VVVVVV{BLK}M"
EG 290 PRINT"{2 RIGHT}N{YEL}VV
       VVVVVVVVVVVVVVVVVVVVVV
       VVVVVVVV{BLK}M"
GJ 300 PRINT"{2 RIGHT}O[Y]O
       [31 Y]OP{DOWN}{LEFT}[N]
       {UP}"
CB 310 PRINT"{2 RIGHT}[H]{UP}
BM 320 PRINT"{HOME}{9 DOWN}
       {5 RIGHT}[P][8]";:X=31:
       GOSUB70
HQ 330 PRINT"{UP}{4 RIGHT}
       {BLK}N";:X=32:GOSU870
XR 340 PRINT"{3 RIGHT}{BLK}N";
       :X=33:GOSUB70
BQ 350 PRINT"{UP}{3 RIGHT}
       {8LK}{RVS}{34 SPACES}
       {8LU}":IFE=1THEN510
JD 360 V=53248:IFE=0THENPRINT"
       {HOME}{DOWN}{10 RIGHT}P
       LEASE WAIT A MOMENT"
HX 370 S1=12288:S2=12352:S3=12
       416:S4=12480:S5=12544
AQ 380 FORX=0TO41
DJ 390 READQ1:POKES1+X,Q1:POKE
       S2+X,Q1:POKES3+X,Q1:POK
       ES4+X,Q1:POKES5+X,Q1
RG 400 NEXTX
KF 410 FORS1=12330TO12350:READ
       Q1:POKES1,Q1:NEXT
EH 420 FORS2=12394TO12414:READ
       Q1:POKES2,Q1:NEXT
8G 430 FORS3=12458TO12478:READ
       Q1:POKES3,Q1:NEXT
AE 440 FORS4=12522TO12542:READ
       Q1:POKES4,Q1:NEXT
CH 450 FORS5=12586TO13182:READ
       Q1:POKES5,Q1:NEXT
XF 460 S6=13183:S7=13247
HQ 470 FORXX=0TO45:READQ1:POKE
       S6+XX,Q1:POKES7+XX,Q1:N
       EXT
PE 480 FORS6=13229TO13246:READ
       Q1:POKES6,Q1:NEXT
HM 490 FORS7=13293TO13311:READ
       Q1:POKES7,Q1:NEXT
EM 500 IFE=0THENPRINT"{HOME}
       {DOWN}{10 RIGHT}
       {20 SPACES}"
HE 510 T2=TI/60:POKE2040,192
BH 520 POKEV+39,4:POKEV+40,0:P
       OKEV,65:POKEV+2,65:POKE
       V+1,Y2:POKEV+3,69:POKEV
       +16,3
CX 530 IFOA=0THENR9=9:POKEV+21
       ,1
ER 540 IFOA=1THENR9=6:POKEV+21
       ,3
BA 550 P=192
SA 560 FORX=321TO35STEP-3
KB 570 IFW=2ANDD=8THENB=22
RB 580 IFW=2ANDD=9THENB=21
KC 590 C=(C+1)+O
GP 600 GOSUB1330
HC 610 PRINT"{HOME}{RIGHT}SCOR
       E:"C"{LEFT}{4 SPACES}"T
       AB(16)"BONUS"T"{LEFT} "
       TAB(32)"LEVEL"O
FB 620 IFX<256THENPOKEV+16,0:P
       OKEV,X:POKEV+2,X
EF 630 IFX>255THENPOKEV+16,3:P
       OKEV,X-256:POKEV+2,X-25
       6
KE 640 F=0:IFX<295ANOX>50THENG
       OSUB840
RM 650 IF F=1 THEN1120
KB 660 P1=P1+1:IFP1>207THENP1=
       206
KG 670 POKE2041,P1
BR 680 POKE2040,P
JO 690 P=P+1:IFP>196THENP=192
DE 700 GETA$:IFA$=" "THENGOSUB
       1640
RG 710 JV=PEEK(56320)
HP 720 JV=15-(JVANO15)
RA 730 IFJV=4THENGOSUB870
JX 740 IFJV=8THENGOSUB990
```

```
GR 750 IF F=1 THEN1120
FR 760 NEXTX
CB 770 C=C+T:T=0
JK 780 IFDA=1THENDA=0:Y2=69:GO
       TO800
JS 790 DA=1:Y2=62
FC 800 IFD<9THENB=B-1:D=D+1
XF 810 IFW=2ANDD<9THENB=B-7
AS 820 IFD=9THENPX=200:PZ=204
PG 830 GOTO510
QR 840 R=INT(R9*RND(1))+1
EJ 850 IFR>2THENRETURN
GE 860 IFR=1THEN990
CC 870 P=197:POKE2040,P
BS 880 IFW=1THENGOSUB1930
DX 890 IFW=2THENGOSUB1730
KA 900 IFM1=1THENJV=8
RR 910 IPM1=2THENJV=4
SQ 920 IFJV=8THENP=P-1:POKE204
       0,P:IFP<197THENRETURN
FB 930 IFJV=8THEN880
PE 940 P=P+1:POKE2040,P
PB 950 GOSUB1330
DH 960 PRINT"{HOME}"TAB(21)T
RS 970 IFP>196ANDP<PXTHEN880
BR 980 F=1:RETURN
GB 990 P=201:POKE2040,P
EK 1000 IPW=1THENGOSUB1930
MX 1010 IFW=2THENGOSUB1730
CS 1020 IPM1=1THENJV=4
AA 1030 IPM1=2THENJV=8
MH 1040 IPJV=4THENP=P-1
QG 1050 GOSUB1330
BH 1060 PRINT"{HOME}"TAB(21)T
MQ 1070 IFP<201THENP=196:RETUR
       N
BD 1080 IFJV=4THENPOKE2040,P:G
       OTO1000
PX 1090 P=P+1:POKE2040,P
QQ 1100 IPP>200ANDP<PZTHEN1000
JS 1110 F=1:RETURN
HD 1120 Z2=69:P=205:U7=200
DF 1130 POKEU1,15:POKEU2,0:POK
       EU3,247:POKEU4,17
PJ 1140 FORZ=Z2TO207STEP8
QR 1150 U7=U7-8
RQ 1160 POKEU5,U7
BJ 1170 POKEV+1,Z
CA 1180 POKE2040,P:NEXTZ
PX 1190 Z2=Z2+30
JJ 1200 FORX=207TOZ2STEP-8
SX 1210 U7=U7+8
EB 1220 POKEU5,U7
RJ 1230 POKEV+1,X:POKE2040,P:N
       EXTX
8C 1240 IFZ2<236THEN1140
PF 1250 POKE54276,16
BF 1260 POKE53269,0:PRINT"
       {DOWN}{7 RIGHT}{YEL}PR
       ESS {RVS}RETURN{OFF} T
       O PLAY AGAIN"
FS 1270 PRINT"{DOWN} PRESS SPA
       CE BAR OR PIREBUTTON P
       OR MENU"
SS 1280 GETA$:JV=PEEK(56320):F
       R=JVAND16
GH 1290 IP(A$=""OR(A$<>" "ANDA
       $<>CHR$(13)))ANDFR<>0T
       HEN1280
SA 1300 C=0:M1=0:AV=0:PX=201:P
       Z=205:E=1
CP 1310 IFA$=CHR$(13)THEN250
BA 1320 DA=0:GOTO80
XA 1330 T3=3*INT(TI/60-T2):T=1
       000-T3:IFT<0THENT=0
QF 1340 RETURN
RS 1350 DATA0,240,0,1,240,0,3,
       176,0,3,48,0,3,48,0,2,
       24,0,12,8,0
PG 1360 DATA0,120,0,0,248,0,1,
       216,0,1,140,0,1,134,0,
       0,131,0,3,130,0,0
JC 1370 DATA0,0,0,0,0,0,0,28,0
       ,64,46,0,48,126,0,28,4
       6,0,7,24,0,1,254,0,0,8
       7,128
AH 1380 DATA0,108,224,0,124,56
       ,0,68,4,0,124,0,0,124,
       0,0,248,0,1,216,0,3,24
       ,0
CP 1390 DATA6,24,0,28,24,0,0,2
       4,0,0,48,0,0
PC 1400 DATA0,0,0,0,0,0,32,24,
       0,16,44,0,8,124,0,4,44
       ,0,3,16,0,1,252,0,0,17
       4,0
AS 1410 DATA0,219,0,0,248,128,
       0,136,64,0,248,32,0,24
       8,0,1,240,0,7,48,0,28,
       48,0
SD 1420 DATA112,48,0,128,48,0,
       0,48,0,0,96,0,0
AQ 1430 DATA0,32,0,0,64,0,0,15
       6,0,1,46,0,1,126,0,1,4
       4,0,1,152,0,1,25,0,0,1
       74,0
QK 1440 DATA0,219,0,0,184,128,
       1,208,64,1,224,32,131,
       240,0,255,240,0,0,48,0
GC 1450 DATA0,48,0,0,48,0,0,48
       ,0,0,48,0,0,96,0,0
GJ 1460 DATA0,8,0,0,16,0,0,39,
       16,0,75,160,0,95,192,0
       ,75,128,0,127,128,64,5
       4,0,96
CR 1470 DATA46,0,56,92,0,15,23
       2,0,1,252,0,0,60,0,0,6
       0,0,0,56,0,0,56,0,0,48
       ,0,0,48
GE 1480 DATA0,0,16,0,0,24,0,0,
       48,0,0
DM 1490 DATA0,0,0,0,0,0,0,112,
       0,0,184,8,1,248,48,0,1
       84,192,0,115,0,1,252,0
       ,6,168,0
ME 1500 DATA24,216,0,96,248,0,
       128,136,0,0,248,0,0,24
       8,0,0,248,0,0,216,0,0,
       204,0
KG 1510 DATA0,198,0,0,195,0,0,
       193,192,3,128,0,0
GK 1520 DATA0,0,0,0,0,16,0,112
       ,32,0,184,64,1,248,128
       ,0,185,0,0,114,0
AJ 1530 DATA1,252,0,2,168,0,4,
       216,0,8,248,0,16,36,0,
       96,248,0,0,248,0,0,248
       ,0
FC 1540 DATA0,204,0,0,198,0,0,
       195,0,0,193,224,0,192,
       192,3,128,0,0
DK 1550 DATA0,8,0,0,4,0,0,226,
       0,1,113,0,3,241,0,1,11
       3,0,0,226,0,1,252,0
CF 1560 DATA3,80,0,7,176,0,13,
       240,0,B,144,0,16,240,0
       ,96,248,0,0,252,8
PB 1570 DATA0,207,240,0,192,0,
       0,192,0,0,192,0,0,192,
       0,1,128,0,0
EX 1580 DATA0,64,0,16,32,0,39,
       176,0,37,208,0,47,208,
       0,47,208,0,37,208,64
CA 1590 DATA19,144,48,8,240,32
       ,7,208,96,3,176,192,0,
       243,128,0,158,0,0,252,
       0
HB 1600 DATA0,240,0,1,128,0,1,
       128,0,1,128,0,0,192,0,
       0,192,0,1,0,0,0
JE 1610 DATA0,0,128,0,0,136,4,
       1,152,4,67,48,4,67,96,
       4,71,96,4,199,96
BA 1620 DATA36,135,96,123,135,
       96,181,135,96,255,255,
       224,119,247,224,3,255,
       192
AG 1630 DATA0,0,0,0,0,0,0,0,0,0,
       0,0,0,0,0,0,0,0,0,0,0,
       0,0,0,0,0,0
CC 1640 T4=INT(TI/60)
CS 1650 GETA$:IFA$=""THEN1650
CA 1660 T5=INT(TI/60)
K8 1670 T2=T2+T5-T4:RETURN
QE 1680 DATA0,0,0,0,0,0,0,0,0,
       0,0,0,0,0,0,0,0
CD 1690 DATA2,2,0,1,4,0,0,248,
       0,0,32,0,0,32,0,0,32,0
       ,0,32,0
DA 1700 DATA0,32,0,0,32,0,0,24
       8,0,1,36,0,2,34,0,3,25
       4,0,2,34,0,1,36,0
DQ 1710 DATA0,248,0
CX 1720 DATA1,140,0,2,82,0,2,3
       4,0,2,82,0,1,140,0,0,2
       48,0,0
RK 1730 R1=INT(26*RND(1))+1
RS 1740 X9=INT(X/8)-4
KS 1750 R2=R1+64
JS 1760 A=0
KF 1770 PRINT"{HOME}{2 DOWN}"T
       AB(X9)CHR$(R2)
HC 1780 GETA$:A=A+1
PB 1790 IFA=BTHENM1=2:GOSUB190
       0:GOTO1830
FD 1800 IFA$=""THEN1780
SF 1810 IFA$=CHR$(R2)THENM1=1:
       GOSUB1870:GOTO1830
SB 1820 M1=2:GOSUB1840
QC 1830 PRINT"{HOME}{2 DOWN}"T
       AB(X9)" ":RETURN
SJ 1840 POKEU1,15:POKEU2,45:PO
       KEU3,165:POKEU4,33
AB 1850 POKEU5,6:POKEU6,5
XJ 1860 FORT=1TO200:NEXT:POKEU
       4,32:POKEU5,0:POKEU6,0
       :RETURN
QE 1870 POKEU1,15:POKEU2,0:POK
       EU3,247:POKEU4,17
CX 1880 POKEU5,40:POKEU6,0
KP 1890 FORT=1TO100:NEXT:POKEU
       4,16:RETURN
ME 1900 POKEU1,15:POKEU4,33:PO
       KEU2,15
EH 1910 FORT=50TO5STEP-2:POKEU
       5,16:POKEU6,T:NEXT
B8 1920 POKEU4,0:RETURN
SB 1930 JV=PEEK(56320)
SE 1940 GETA$:IFA$<>""THEN1940
XS 1950 JV=15-(JVAND15)
SJ 1960 IFJV=4ORJV=8THENAV=0:R
       ETURN
AF 1970 AV=AV+1:IFAV=BTHENAV=0
       :RETURN
GD 1980 GOTO1930
```

Program 2: Atari Tightrope

Version by Kevin Mykytyn, Editorial Programmer

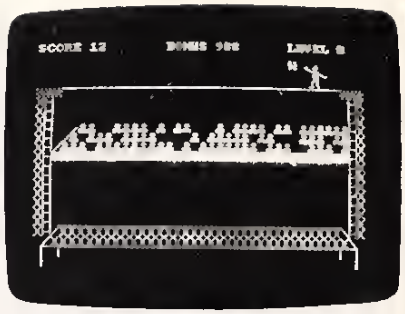
```
CI 10 POKE 106,96:GRAPHICS 7
      :C=0:GRAPHICS 0:POKE 7
      52,1:POBITION 14,10:PR
      INT "PLEASE WAIT"
OF 20 FOR A=30720 TO 31B41:R
      EAD B:C=C+B:POKE A,B:N
      EXT A:IF C<>70649 THEN
       PRINT "DATA ERROR":ST
      OP
AP 30 DIM P$(32),K$(1),A$(1)
      :OPEN #4,4,4,"K:"
GN 40 GRAPHICS 0:POKE 752,1:
      POKE 710,15:POKE 709,0
      :POSITION 14,9:PRINT "
      (1) GAME":POSITION 14,
      11:PRINT "(2) TYPING"
ID 50 GET #4,K:K$=CHR$(K):IF
```



"Tightrope" for Atari 400, 800, XL, and XE computers.

```
      K$<>"1" AND K$<>"2" T
     HEN 50
JN 60  W=VAL(K$)
PE 70  PRINT "{3 DOWN}
     {3 SPACES}ENTER LEVEL
     OF DIFFICULTY (0-9)"
JJ 80  GET #4,K:K$=CHR$(K)
HJ 90  IF K$<"0" OR K$>"9" TH
     EN 80
DL 100  B=VAL(K$):O=8:B=12-B
OF 110  IF W=2 THEN B=B*4
EC 120  P$="0110011011111111102
     2002202222222":GRAP
     HICS 7:OL=PEEK(560)+2
     56*PEEK(561):POKE OL+
     6,2
NI 130  SETCOLOR 0,5,5:SETCOL
     OR 1,13,12:POKE 710,0
     :SCREEN=PEEK(88)+256*
     PEEK(89)+40
KP 140  COLOR 1:FOR A=8 TO 14
     4 STEP 136:FOR Q=0 TO
      8 STEP 8:PLOT A+Q,16
     :DRAWTO A+Q,80:NEXT Q
     :POKE 752,1
JN 150  FOR C=16 TO 80 STEP 4
     :PLOT A,C:DRAWTO A+8,
     C:NEXT C:NEXT A
NL 160  POKE 54279,112:POKE 5
     3277,3:POKE 559,62:PO
     KE 623,1:FOR A=704 TO
      706:POKE A,78:NEXT A
KG 170  COLOR 2:PLOT 4,16:DRA
     WTO 156,16:COLOR 1:PL
     OT 18,72:DRAWTO 142,7
     2:DRAWTO 132,64:DRAWT
     O 28,64:DRAWTO 18,72
NJ 180  COLOR 2:FOR A=30 TO 1
     32 STEP 6:PLOT A,65:D
     RAWTO A-7,71:PLOT A,6
     5:DRAWTO A+7,71:NEXT
     A
NE 190  PLOT 26,67:DRAWTO 28,
     71:PLOT 135,68:DRAWTO
      132,71
AE 200  COLOR 1:PLOT 18,73:DR
     AWTO 18,80:PLOT 142,7
     3:DRAWTO 142,80:PLOT
     28,73:DRAWTO 28,76:PL
     OT 132,73:DRAWTO 132,
     76
HB 210  COLOR 2:PLOT 18,44:DR
     AWTO 143,44:DRAWTO 14
     3,48:DRAWTO 18,48:DRA
     WTO 18,44:DRAWTO 30,3
     1:DRAWTO 143,31
OD 220  C=0:FOR Y=40 TO 32 ST
     EP -4:FOR X=23+5*C TO
      142 STEP 5:GOSUB 240
     :NEXT X:C=C+1:NEXT Y
AE 230  PX=6:PZ=11:Y2=44:P1=1
     3:GOTO 260
LL 240  Q=INT(RND(1)*3):IF Q=
     2 THEN RETURN
CN 250  FOR A=0 TO 3:POSITION
      X,Y+A:PRINT #6;P$(Q*
     16+A*4+1,Q*16+A*4+4):
     NEXT A:RETURN
AL 260  GOSUB 1120:T2=TI/60
PG 270  POKE 205,0:P=0:POKE 2
     06,205:POKE 207,Y2:PO
     KE 209,116:A=USR(3072
     0)
FN 280  FOR X=185 TO 48 STEP
     -1
IO 290  IF W=2 AND O=8 THEN 8
     =22
IG 300  IF W=2 AND O=9 THEN 8
     =21
CC 310  C=C+1+O
OG 320  GOSUB 1120:T3=3*(INT(
     TI/60-T2)):T=1000-T3
NF 330  IF T<0 THEN T=0
OB 340  POKE 656,1:POKE 657,1
     :PRINT "SCORE: ";C;:P
     OKE 657,16:PRINT "BON
     US: ";T;" ";:POKE 657
     ,32:PRINT "LEVEL: ";O
FP 350  POKE 205,P:POKE 206,X
EM 360  P=P+1:IF P>2 THEN P=0
MB 370  IF X<175 AND X>50 THE
     N GOSUB 460
CN 380  IF PEEK(764)=33 THEN
     GOSUB 910
OK 390  IF STICK(0)=7 THEN GO
     SUB 490
AJ 400  IF STICK(0)=11 THEN G
     OSUB 630
CM 410  NEXT X
NO 420  C=C+T:T=0
II 430  IF O<9 THEN 8=8-1:O=O
     +1
MJ 440  IF W=2 AND O<9 THEN 8
     =8-3
GK 450  GOTO 260
LM 460  R=INT(29*RND(1))+1
PL 470  IF R>2 THEN RETURN
LD 480  IF R=1 THEN 630
NJ 490  P=3:POKE 205,P
GB 500  IF W=1 THEN GOSUB 108
     0
OI 510  IF W=2 THEN GOSUB 950
IA 520  IF M1=1 THEN JV=11
FM 530  IF M1=2 THEN JV=7
OE 540  IF JV=11 THEN P=P-1:P
     OKE 205,P:IF P<3 THEN
      RETURN
CM 550  IF JV=1I THEN 500
FA 560  P=P+1:POKE 205,P
JM 570  GOSUB 1I20:T3=3*INT(T
     I/60-T2):T=I000-T3
NN 580  IF T<1 THEN T=0
AA 590  POKE 656,1:POKE 657,2
     2:PRINT " ";T
LA 600  IF P>2 AND P<PX THEN
     500
EO 610  POKE 205,12:GOSUB 760
MI 620  RETURN
NK 630  P=8:POKE 205,P
GG 640  IF W=1 THEN G08U8 108
     0
DM 650  IF W=2 THEN G08U8 950
FK 660  IF M1=1 THEN JV=7
IM 670  IF M1=2 THEN JV=I1
KL 680  IF JV=7 THEN P=P-1
PA 690  GOSUB 1120:T3=3*(INT(
     TI/60-T2)):T=1000-T3
NH 700  IF T<1 THEN T=0
PK 710  POKE 656,I:POKE 657,2
     2:PRINT " ";T
PE 720  IF P<8 THEN P=2:RETUR
     N
LL 730  IF JV=7 THEN POKE 205
     ,P:GOTO 640
FA 740  P=P+1:POKE 205,P
LN 750  IF P>2 AND P<PZ THEN
     640
IN 760  Z2=44:P=12:U7=200
FI 770  FOR Z=Z2 TO 158 STEP
     3
PO 780  SOUND 1,Z,10,I5
DN 790  POKE 205,P:POKE 207,Z
     :NEXT Z
HL 800  Z2=Z2+30
MO 810  FOR X=158 TO Z2 STEP
     -3
PN 820  SOUND 1,X,10,15
OE 830  POKE 207,X:POKE 205,P
     :NEXT X
FF 840  IF Z2<140 THEN 770
MC 850  SOUND 1,0,0,0:POKE 65
     6,2:POKE 657,5:PRINT
     "  PRESS RETURN TO PL
     AY AGAIN":POKE 764,25
     5
OM 860  PRINT "PRESS SPACE BA
     R OR FIREBUTTON FOR M
     ENU";
GL 870  IF PEEK(764)=12 THEN
     POKE 206,0:E=1:AV=0:M
     1=0:C=0:PX=201:PZ=205
     :PRINT "{CLEAR}":GOTO
      230
ML 880  IF PEEK(764)<>33 AND
     STRIG(0)<>0 THEN 870
GI 890  POKE 206,0:E=1:C=0:M1
     =0:AV=0:PX=201:PZ=205
OG 900  GOTO 40
BN 910  POKE 764,255:GOSUB 11
     20:T4=INT(TI/60)
OP 920  GET #4,A
EO 930  GOSUB 1120:T5=INT(TI/
     60)
GJ 940  T2=T2+T5-T4:RETURN
OO 950  R1=INT(26*RND(1))+1
EO 960  X9=INT((X-48)/4)
HE 970  R2=R1+32
EP 980  A=0
CN 990  POKE 764,255:K=255:PO
     KE SCREEN+X9,R2
PJ I000  IF PEEK(764)<>255 TH
     EN GET #4,K:K=K-32
DM 1010  A=A+1:IF A=8 THEN M1
     =2:GOSUB 1060:GOTO 1
     050
GG 1020  IF K=255 THEN 1000
DF 1030  IF K=R2 THEN M1=1:GO
     SUB 1070:GOTO 1050
OO 1040  M1=2:GOSUB 1060
OM 1050  POKE SCREEN+X9,0:RET
     URN
LO 1060  FOR V=15 TO 0 STEP -
     1:SOUND 1,200,10,V:N
     EXT V:RETURN
JO 1070  FOR V=15 TO 0 8TEP -
     1:SOUNO 1,60,10,V:NE
     XT V:RETURN
KF 1080  JV=STICK(0)
JK 1090  IF JV=7 OR JV=11 THE
     N AV=0:RETURN
LF 1100  AV=AV+1:IF AV=8 THEN
      AV=0:RETURN
MF 1110  GOTO 1080
JP 1120  TI=PEEK(18)*65536+PE
     EK(19)*256+PEEK(20):
     RETURN
CK 1130  OATA 169,0,I33,186,1
     65,209,133,187,162,3
     ,160,0,152,145,186,2
     00,208,251,230,187,2
     02,I6,246,160,34,162
NP 1140  OATA 120,169,7,32,92
     ,228,I04,96,216,169,
     0,133,77,32,45,120,7
     6,98,228,I65,206,141
     ,0,208,24,105
HN I150  OATA 8,141,I,208,24,
     105,8,141,2,208,165,
     205,133,203,169,0,13
     3,204,162,6,6,203,38
     ,204,202,240
HN I160  OATA 17,224,3,208,24
     5,165,203,141,184,12
     0,165,204,141,185,12
     0,76,72,120,165,203,
     24,109,184,I20,I33,2
```

```
        Ø3
BN 1170 OATA 165,204,109,185
        ,120,133,204,165,203
        ,24,105,186,141,146,
        120,165,204,105,120,
        141,147,120,165,209,
        133,204
FO 1180 OATA 169,3,133,208,1
        69,Ø,133,203,164,207
        ,145,203,200,162,Ø,1
        89,255,255,145,203,2
        00,232,224,24,208,24
        5
HF 1190 DATA 169,Ø,145,203,1
        73,146,120,24,105,24
        ,141,146,120,173,147
        ,120,105,Ø,141,147,1
        20,230,204,198,208,2
        08
CN 1200 DATA 207,96,Ø,Ø,Ø,Ø,
        Ø,Ø,Ø,Ø,96,63,Ø,Ø,Ø,
        Ø,Ø,Ø,Ø,Ø,Ø,Ø,1,1,1,
        1
EC 1210 DATA 1,3,56,60,124,5
        6,48,Ø,124,255,254,1
        24,124,124,124,124,1
        24,124,124,204,134,1
        34,131,131,129,3
MM 1220 DATA Ø,Ø,Ø,Ø,Ø,Ø,12,
        248,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,
        Ø,Ø
OP 1230 DATA Ø,Ø,Ø,Ø,128,Ø,Ø
        ,Ø,Ø,Ø,Ø,Ø,96,63,Ø,Ø
        ,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
MF 1240 DATA Ø,Ø,Ø,Ø,56,60,1
        24,56,48,Ø,124,255,2
        54,124,124,124,124,1
        24,124,124,124,108,1
        08,198,195,97
JN 1250 DATA 96,193,Ø,Ø,Ø,Ø,
        Ø,Ø,12,248,Ø,Ø,Ø,Ø,Ø
        ,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,128
        ,192,128
PI 1260 DATA Ø,Ø,Ø,Ø,Ø,Ø,Ø,96,
        63,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
        ,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,56,60
HI 1270 DATA 124,56,48,Ø,124
        ,255,254,124,124,124
        ,124,124,124,124,124
        ,108,120,120,60,54,2
        8,48,Ø,Ø,Ø,Ø
LC 1280 DATA Ø,Ø,12,248,Ø,Ø,
        Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,
        Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
KD 1290 OATA Ø,96,63,Ø,Ø,Ø,Ø
        ,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
        ,Ø,56,60,124,56,48,Ø
        ,127,254
OK 1300 DATA 252,124,124,124
        ,124,124,124,124,124
        ,108,108,198,195,97,
        97,227,Ø,Ø,Ø,Ø,Ø,12,
        248,Ø,Ø,Ø
JC 1310 OATA Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
        ,Ø,Ø,Ø,128,128,Ø,Ø,Ø
        ,Ø,Ø,Ø,Ø,Ø,Ø,Ø,1,15,56
        ,96
AM 1320 DATA Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
        ,Ø,Ø,Ø,Ø,Ø,56,60,124,5
        6,48,1
CB 1330 DATA 127,254,252,124
        ,124,124,124,124,124
        ,124,124,108,108,198
        ,195,97,99,192,Ø,Ø,Ø
        ,12,120,192,Ø,Ø
HN 1340 OATA Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
        ,Ø,Ø,Ø,Ø,Ø,128,Ø,Ø,Ø
        ,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,1,3
EB 1350 DATA 6,12,56,Ø,Ø,Ø,Ø,Ø
        ,Ø,Ø,Ø,Ø,Ø,Ø,Ø,56,60
        ,124,56,48,1,127,254
        ,252,124,124,124
FK 1360 DATA 124,124,124,124
        ,124,108,108,204,198
        ,99,102,192,Ø,28,48,
        96,192,128,Ø,Ø,Ø,Ø,Ø
        ,Ø,Ø,Ø
NC 1370 OATA Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
        ,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
        ,1,15,56,96,Ø,Ø,Ø,Ø
EB 1380 DATA Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
        ,56,60,124,56,48,1,1
        27,254,252,124,124,1
        24,124,124,124,124,1
        24,108
EN 1390 OATA 108,198,195,97,
        99,192,Ø,Ø,Ø,12,120,
        192,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,
        Ø,Ø,Ø,Ø,Ø,Ø,Ø
DD 1400 OATA Ø,128,Ø,Ø,Ø,Ø,Ø,Ø
        ,Ø,Ø,Ø,96,63,Ø,Ø,Ø,Ø
        ,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
HN 1410 DATA Ø,Ø,56,60,124,5
        6,48,Ø,124,255,254,1
        24,124,124,124,124,1
        24,124,124,108,108,1
        98,195,97,96,193
MD 1420 DATA Ø,Ø,Ø,Ø,Ø,Ø,Ø,12,
        248,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,
        Ø,Ø
NI 1430 DATA Ø,Ø,Ø,128,192,1
        28,Ø,Ø,Ø,Ø,Ø,96,63,Ø
        ,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
        ,Ø,Ø
NP 1440 OATA Ø,Ø,Ø,Ø,56,60,1
        24,56,48,Ø,248,252,1
        27,124,124,124,124,1
        24,124,124,124,108,1
        08,198,198,99
GD 1450 DATA 67,199,Ø,Ø,Ø,Ø,
        Ø,Ø,Ø,12,248,Ø,Ø,Ø,Ø
        ,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
        ,Ø
DE 1460 DATA Ø,Ø,Ø,96,56,14,
        3,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,
        Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,56,6
        Ø
DN 1470 OATA 124,56,48,Ø,248
        ,252,127,125,124,124
        ,124,124,124,124,124
        ,108,108,204,198,102
        ,195,6,Ø,Ø,Ø,Ø
FL 1480 DATA Ø,Ø,Ø,Ø,Ø,224,5
        6,12,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
        ,Ø,Ø,Ø,Ø,Ø,Ø,Ø,24,12,6
        ,3
DD 1490 OATA 1,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
        ,Ø,Ø,Ø,Ø,Ø,Ø,Ø,1,Ø,Ø
        ,56,60,124,56,48,Ø,2
        48,254
BI 1500 DATA 127,125,124,124
        ,124,124,124,124,124
        ,108,108,204,108,198
        ,6,14,Ø,Ø,Ø,Ø,Ø,Ø,Ø,
        Ø,Ø,128
BA 1510 DATA 192,96,48,Ø,Ø,Ø
        ,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,6,6
        ,6,3,3,3,1,Ø,Ø,Ø,Ø,Ø
II 1520 OATA Ø,Ø,Ø,Ø,Ø,Ø,1,1
        ,1,1,1,3,56,60,124,5
        7,49,1
KL 1530 OATA 255,254,254,124
        ,124,124,124,124,124
        ,124,124,204,140,140
        ,134,134,134,14,192,
        192,192,128,128,128,
        Ø,Ø
CI 1540 DATA Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
        ,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
        ,3,7,3,1,Ø,1,3,6
CK 1550 DATA 28,Ø,32,48,159,
        192,127,Ø,Ø,Ø,Ø,Ø,Ø,
        Ø,Ø,Ø,128,192,192,Ø,
        96,240,248,124,60,60
LN 1560 OATA 60,60,252,124,2
        48,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
        ,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
        ,Ø,Ø
FD 1570 OATA Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
        ,Ø,Ø
```



IBM PC/PCjr version of "Tightrope."

Program 3: IBM PC/PCjr Tightrope

Version by Patrick Parrish, Programming Supervisor

```
HG 10 GOTO 120
NO 20 PUT (X,Y),W3,PSET:RETURN
MM 30 PUT (X,Y),W2,PSET:RETURN
LC 40 PUT (X,Y),W1,PSET:RETURN
CJ 50 PUT (X,Y),L1,PSET:RETURN
DJ 60 PUT (X,Y),L2,PSET:RETURN
EJ 70 PUT (X,Y),L3,PSET:RETURN
HN 80 PUT (X,Y),R1,PSET:RETURN
IN 90 PUT (X,Y),R2,PSET:RETURN
LB 100 PUT (X,Y),R3,PSET:RETURN
BO 110 PUT (X,Y),W4,PSET:RETURN
OE 120 KEY OFF:WIDTH 40:DEF SEG=
       0:POKE 1047,PEEK(1047) OR
        64:SCREEN 1:COLOR ,0:CLS
       :LOCATE 12,15,0:PRINT "PL
       EASE WAIT":GOSUB 1030:GOT
       D 240
BP 130 JV=0:A$=INKEY$:IF A$=CHR$
       (47) THEN JV=1:AV=0:RETUR
       N ELSE IF A$=CHR$(90) THE
       N JV=2:AV=0:RETURN
LN 140 AV=AV+1:IF AV=8*2 THEN AV
       =0:RETURN
CH 150 GOTO 130
KO 160 R1=INT(26*RND(1))+1:X9=IN
       T(X/8):R2=R1+64:A=0:LOCAT
       E 3,X9:PRINT CHR$(R2)
HH 170 A$=INKEY$:A=A+1:IF A=8 TH
       EN M1=2:GOSUB 1010:GOTO 2
       10
HL 180 IF A$="" THEN 170
FF 190 IF A$=CHR$(R2) THEN M1=1:
       GOSUB 1000:GOTO 210
CO.200 M1=2:GOSUB 990
CN 210 LOCATE 3,X9:PRINT " ":RET
       URN
FO 220 T3=3*INT(TIMER-T2):T=1000
       -T3:IF T<0 THEN T=0
HC 230 RETURN
OH 240 RANDOMIZE TIMER:CLS:LOCAT
       E 11,15:PRINT "1- GAME":L
       OCATE 13,15:PRINT "2- TYP
       ING"
EO 250 A$=INKEY$:IF A$="" THEN 2
       50
IB 260 W=VAL(A$):IF W<1 OR W>2 T
       HEN 250
KK 270 LOCATE 17,5:PRINT "ENTER
       LEVEL OF DIFFICULTY (0-9)
       "
DD 280 A$=INKEY$:IF A$="" OR (A$
       <"0" OR A$>"9") THEN 280
       ELSE B=VAL(A$)
HK 290 O=B:B=10-B:IF W=1 THEN B=
       B*2
DH 300 IF W=2 THEN B=B*50
KN 310 CLS:FOR X=0 TO 36 STEP 36
       :FOR J=0 TO 32:PUT (X*8+J*
       8,40),S14:NEXT J,X:LINE (
       24,40)-(287,40),3
LP 320 FOR J=0 TO 32 STEP 32:FOR
        I=1 TO 15:PUT (J*8,I*8+4
       0),S15:PUT (J*8+8+(J=32)*
       16,I*8+40),S16:NEXT I,J
```

```
FC 330 'LINE (40,74)-(8*37-1,74)
       :LINE (39,75)-(8*37-1,75)
BF 340 FOR R=1 TO 3:FOR C=6-R TO
        36:X=RND(1):ROW=64+8*R:C
       OL=C*8:IF X<.3 THEN PUT(C
       OL,ROW),S17:GOTO 380
IP 350 IF X>=.3 AND X<.6 THEN PU
       T(COL,RDW),S18:GOTO 380
BP 360 IF X>=.6 AND X<.9 THEN PU
       T(COL,RDW),S19:GOTO 380
DK 370 LOCATE ROW/8,COL/8:PRINT
       " "
GJ 380 NEXT C,R
HN 390 LINE (39,75)-(17,96):LINE
        (38,75)-(16,96):FOR I=96
        TO 96+7:LINE (16,I)-(8*3
       7-1,I):NEXT I
FN 400 LINE (23,20*8-1)-(36*8,20
       *8-1):LINE (7,22*8)-(38*8
       ,22*8):LINE (23,20*8)-(7,
       22*8):LINE (36*8,20*8-1)-
       (38*8,22*8)
KG 410 FOR R=0 TO 1:FOR C=3-R TO
        35+R:PUT (C*8,(R+20)*8),
       S15:NEXT C,R
IC 420 LINE (7,22*8)-(7,24*8):LI
       NE (38*8,22*8)-(38*8,24*8
       ):LINE (23,22*8)-(23,23*8
       ):LINE (36*8,22*8)-(36*8,
       23*8)
LN 430 Y=19:PX=7:PZ=10:C=0
DJ 440 T2=TIMER
CB 450 IF W=2 AND D=8 THEN 8=60
       ELSE IF W=2 AND D=9 THEN
       8=45
DL 460 P=0:GOSUB 1020:FOR X=288
       TO 4 STEP -3
NL 470 C=C+D+1:GOSUB 220
FK 480 LOCATE 1,1:PRINT "SCORE"C
       "     "TAB(16)"BONUS"T" "T
       AB(31)"LEVEL"D
MC 490 F=0:IF X<268 AND X>24 THE
       N GOSUB 620
BF 500 IF F=1 THEN 840
BD 510 P=P+1:IF P=5 THEN P=1
LE 520 ON P GOSUB 20,30,40,30
IF 530 JV=0:A$=INKEY$:IF A$=" "
       THEN GOSUB 960 ELSE IF A$
       =CHR$(47) THEN JV=1:GOSUB
        640
HH 540 IF A$=CHR$(90) THEN JV=2:
       GOSUB 730
CP 550 IF F=1 THEN 840
GG 560 NEXT X:X=X+3:GOSUB 1020:G
       OSUB 110
NN 570 C=C+T:T=0
KJ 580 IF D<9 THEN 8=8-2:D=D+1
OP 590 IF W=2 AND D<9 THEN 8=8-2
       5
KG 600 IF D=9 THEN PX=6:PZ=9
FN 610 GOTO 440
JC 620 R=INT(9*RND(1))+1:IF R>2
       THEN RETURN
GJ 630 IF R=1 THEN 730
BI 640 P=4:PUT (X,Y),L1,PSET
FF 650 IF W=1 THEN GOSUB 130 ELS
       E IF W=2 THEN GOSUB 160
QL 660 IF M1=1 THEN JV=2 ELSE IF
        M1=2 THEN JV=1
BI 670 IF JV=2 THEN P=P-1:ON P G
       OSUB ,,40,50,60,70:IF P<4
        THEN RETURN ELSE 650
II 680 P=P+1:ON P GOSUB ,,,50,60
       ,70
GC 690 GOSUB 220
HK 700 LOCATE 1,21:PRINT T
PD 710 IF P>3 AND P<PX THEN 650
EI 720 F=1:RETURN
JD 730 P=7:PUT (X,Y),R1,PSET
FE 740 IF W=1 THEN GOSUB 130 ELS
       E IF W=2 THEN GOSUB 160
BB 750 IF M1=1 THEN JV=1 ELSE IF
        M1=2 THEN JV=2
IE 760 IF JV=1 THEN P=P-1
GP 770 GOSUB 220
IK 780 LOCATE 1,21:PRINT T
NB 790 IF P<7 THEN P=3:RETURN
DJ 800 IF JV=1 THEN ON P GOSUB ,
       ,,,,,80,90,100:GOTO 740
BH 810 P=P+1:ON P GOSUB ,,,,,,80
       ,90,100
DE 820 IF P>6 AND P<PZ THEN 740
EL 830 F=1:RETURN
NN 840 P=10:PUT (X,Y),W4,PSET:RE
       M FALLING MAN
KF 850 Z2=23:FOR Z=Z2 TO 150 STE
       P 6:IF Z>Z2 THEN PUT (X,Z
       -6),812
ND 860 PUT (X,Z),S12:SOUND (Z+15
       )*2,.08
PC 870 NEXT Z:PUT (X,Z-6),S12:Y=
       Y+50
HA 880 FOR Z=150 TO Y STEP -6:IF
        Z<150 THEN PUT (X,Z+6),8
       13
BC 890 PUT (X,Z),S13:SOUND (Z+15
       )*2,1:NEXT Z:PUT (X,Z+6),
       S13
EA 900 FOR Z=Y TO 150 STEP 6:IF
       Z>Y THEN PUT (X,Z-6),813
LD 910 PUT (X,Z),S13:SOUND (Z+15
       )*2,1:NEXT Z:PUT (X,Z-6),
       S13:Y=Y+30:IF Y<150 THEN
       880
DM 920 PUT (X,Z-6),S13:LOCATE 3,
       6:PRINT "PRESS <RETURN> T
       O PLAY AGAIN":LOCATE 4,7:
       PRINT "PRESS <SPACE BAR>
       FOR MENU"
NP 930 A$=INKEY$:IF A$<>" " AND
       A$<>CHR$(13) THEN 930
LC 940 C=0:M1=0:AV=0:IF A$=CHR$(
       13) THEN 310
EE 950 GOTO 240
AB 960 T4=INT(TIMER)
FA 970 A$=INKEY$:IF A$="" THEN 9
       70
BB 980 T5=INT(TIMER):T2=T2+T5-T4
       :RETURN
HK 990 SOUND 37,1:RETURN
LB 1000 SOUND 440,1:RETURN
QL 1010 SOUND 2300,1:RETURN
BL 1020 FOR DE=1 TO 400:NEXT:RET
       URN
EN 1030 REM define shapes
FE 1040 DEFINT E,L,R,S,W
HO 1050 RESTORE 1240:READ X,Y:E=
       (4+INT((X+7)/8)*Y)/2:DIM
        W1(E):W1(0)=X:W1(1)=Y:F
       OR I=2 TO E:READ A$:W1(I
       )=VAL("&H"+A$):NEXT
FB 1060 READ X,Y:E=(4+INT((X+7)/
       8)*Y)/2:DIM W2(E):W2(0)=
       X:W2(1)=Y:FOR I=2 TO E:R
       EAD A$:W2(I)=VAL("&H"+A$
       ):NEXT
EE 1070 READ X,Y:E=(4+INT((X+7)/
       8)*Y)/2:DIM W3(E):W3(0)=
       X:W3(1)=Y:FOR I=2 TO E:R
       EAD A$:W3(I)=VAL("&H"+A$
       ):NEXT
NP 1080 E=(4+INT((42+7)/8)*21)/2
       :DIM W4(E):W4(0)=42:W4(1
       )=21:FOR I=2 TO E:W4(I)=
       0:NEXT
PB 1090 READ X,Y:E=(4+INT((X+7)/
       8)*Y)/2:DIM R1(E):R1(0)=
       X:R1(1)=Y:FOR I=2 TO E:R
       EAD A$:R1(I)=VAL("&H"+A$
       ):NEXT
ND 1100 READ X,Y:E=(4+INT((X+7)/
       8)*Y)/2:DIM R2(E):R2(0)=
       X:R2(1)=Y:FOR I=2 TO E:R
       EAD A$:R2(I)=VAL("&H"+A$
       ):NEXT
LB 1110 READ X,Y:E=(4+INT((X+7)/
       8)*Y)/2:DIM R3(E):R3(0)=
       X:R3(1)=Y:FOR I=2 TO E:R
       EAD A$:R3(I)=VAL("&H"+A$
       ):NEXT
HE 1120 READ X,Y:E=(4+INT((X+7)/
       8)*Y)/2:DIM L1(E):L1(0)=
       X:L1(1)=Y:FOR I=2 TO E:R
       EAD A$:L1(I)=VAL("&H"+A$
       ):NEXT
GC 1130 READ X,Y:E=(4+INT((X+7)/
       8)*Y)/2:DIM L2(E):L2(0)=
       X:L2(1)=Y:FOR I=2 TO E:R
       EAD A$:L2(I)=VAL("&H"+A$
       ):NEXT
FA 1140 READ X,Y:E=(4+INT((X+7)/
       8)*Y)/2:DIM L3(E):L3(0)=
       X:L3(1)=Y:FOR I=2 TO E:R
       EAD A$:L3(I)=VAL("&H"+A$
       ):NEXT
DC 1150 READ X,Y:E=(4+INT((X+7)/
       8)*Y)/2:DIM S12(E):S12(0
       )=X:S12(1)=Y:FOR I=2 TO
       E:READ A$:S12(I)=VAL("&H
       "+A$):NEXT
DK 1160 READ X,Y:E=(4+INT((X+7)/
       8)*Y)/2:DIM S13(E):S13(0
       )=X:S13(1)=Y:FOR I=2 TO
       E:READ A$:S13(I)=VAL("&H
       "+A$):NEXT
CC 1170 READ X,Y:E=(4+INT((X+7)/
       8)*Y)/2:DIM S14(E):S14(0
       )=X:S14(1)=Y:FOR I=2 TO
       E:READ A$:S14(I)=VAL("&H
       "+A$):NEXT
CK 1180 READ X,Y:E=(4+INT((X+7)/
       8)*Y)/2:DIM S15(E):S15(0
       )=X:S15(1)=Y:FOR I=2 TO
       E:READ A$:S15(I)=VAL("&H
       "+A$):NEXT
BC 1190 READ X,Y:E=(4+INT((X+7)/
       8)*Y)/2:DIM S16(E):S16(0
       )=X:S16(1)=Y:FOR I=2 TO
       E:READ A$:S16(I)=VAL("&H
       "+A$):NEXT
PD 1200 READ X,Y:E=(4+INT((X+7)/
       8)*Y)/2:DIM S17(E):S17(0
       )=X:S17(1)=Y:FOR I=2 TO
       E:READ A$:S17(I)=VAL("&H
       "+A$):NEXT
OG 1210 READ X,Y:E=(4+INT((X+7)/
       8)*Y)/2:DIM S18(E):S18(0
       )=X:S18(1)=Y:FOR I=2 TO
       E:READ A$:S18(I)=VAL("&H
       "+A$):NEXT
DI 1220 READ X,Y:E=(4+INT((X+7)/
       8)*Y)/2:DIM S19(E):S19(0
       )=X:S19(1)=Y:FOR I=2 TO
       E:READ A$:S19(I)=VAL("&H
       "+A$):NEXT:RETURN
BG 1230 REM W1
PB 1240 DATA 48,21,0,A,0,0,8022,
       0
HK 1250 DATA 0,80AA,0,20,2A,8000
       ,A0,A
BM 1260 DATA 8002,8,802A,8,AA02,
       AAAA,A0,0
LB 1270 DATA 80AA,0,0,80AA,0,0,8
       0AA,0
NE 1280 DATA 0,80AA,0,0,80AA,0,0
       ,80AA
QJ 1290 DATA 0,0,80A2,0,0,80A2,0
       ,0
HD 1300 DATA 80A2,0,0,802A,0,0,8
       00A,0
FJ 1310 DATA 0,A002,0,0,8002,0,0
       ,802A
KF 1320 DATA 0,0
HD 1330 REM W2
PD 1340 DATA 48,21,0,A,0,0,8022,
       0
HM 1350 DATA 0,80AA,0,20,2A,8000
       ,A0,A
BD 1360 DATA 8002,8,802A,8,AA02,
       AAAA,A0,0
LD 1370 DATA 80AA,0,0,80AA,0,0,8
       0AA,0
NG 1380 DATA 0,80AA,0,0,80AA,0,0
       ,80AA
KK 1390 DATA 0,0,80A2,0,200,8082
```

```
        ,Ø,A0Ø
LG 14ØØ DATA 8ØØ2,Ø,A0Ø,8ØØ2,Ø,A
        Ø0,8ØØ2,Ø
CL 141Ø DATA 2ØØ,A08Ø,Ø,2ØØ,2888
        ,Ø,A0Ø,A0ØØ
KH 142Ø DATA Ø,Ø
IA 143Ø REM W3
PF 144Ø DATA 48,21,Ø,A,Ø,Ø,8Ø22,
        Ø
HO 145Ø DATA Ø,8ØAA,Ø,2Ø,2A,8ØØØ
        ,A0,A
BA 146Ø DATA 8ØØ2,8,8Ø2A,8,AAØ2,
        AAAA,A0,Ø
LF 147Ø DATA 8ØAA,Ø,Ø,8ØAA,Ø,Ø,8
        ØAA,Ø
OI 148Ø DATA Ø,8ØAA,Ø,Ø,8ØAA,Ø,Ø
        ,8ØAA
FN 149Ø DATA Ø,Ø,8ØA2,Ø,2ØØ,A08Ø
        ,Ø,A0Ø
JC 15ØØ DATA 28ØØ,Ø,28ØØ,AØØ,Ø,2
        8ØØ,2ØØ,8Ø
BL 151Ø DATA 28ØØ,Ø,AØ,28ØØ,Ø,AØ
        ,A8Ø2,2ØØ
KN 152Ø DATA 8Ø,Ø
QO 153Ø REM L1
DD 154Ø DATA 48,21,Ø,A,Ø,Ø,8Ø22,
        2
EJ 155Ø DATA Ø,8ØAA,8Ø02,Ø,2A,2A
        ,Ø,2ØA
PD 156Ø DATA AØ,Ø,AA2A,Ø,2ØØ,AØA
        A,Ø,2AØØ
GC 157Ø DATA 8ØAA,Ø,AØØ2,8ØAA,Ø,
        AA,8ØAA,Ø
OC 158Ø DATA 2Ø,8ØAA,Ø,Ø,8ØAA,Ø,
        Ø,8ØAA
KO 159Ø DATA Ø,Ø,8ØA2,Ø,2ØØ,8Ø82
        ,Ø,A0Ø
FA 16ØØ DATA 8ØØ2,Ø,AØØ,AØØØ,Ø,A
        Ø0,28ØØ,Ø
AH 161Ø DATA 2ØØ,2888,Ø,2ØØ,A08Ø
        ,Ø,AØØ,Ø
LL 162Ø DATA Ø,Ø
BL 163Ø REM L2
PL 164Ø DATA 48,21,Ø,A,8ØØA,Ø,8Ø
        22,28
OK 165Ø DATA Ø,8ØAA,AØ,Ø,22A,8Ø,
        Ø,AØA
EG 166Ø DATA Ø,Ø,A82A,Ø,2ØØ,AØAA
        ,Ø,AØØ
HJ 167Ø DATA 8ØAA,Ø,28ØØ,8ØAA,Ø,
        AØØØ,8ØAA,Ø
LL 168Ø DATA 8ØØ2,8ØAA,Ø,A,8ØAA,
        Ø,A8,8ØAA
KA 169Ø DATA Ø,Ø,8ØA2,Ø,2ØØ,8Ø82
        ,Ø,AØØ
GK 17ØØ DATA 8ØØ2,Ø,AØØ,AØØØ,Ø,A
        Ø0,2AØØ,AØ
FL 171Ø DATA 2ØØ,8Ø,AØ,2ØØ,8Ø,2Ø
        ,AØØ,Ø
LN 172Ø DATA Ø,Ø
CI 173Ø REM L3
EF 174Ø DATA 42,21,8ØØ2,28,Ø,AØA
        ,A,Ø
GO 175Ø DATA 2228,8Ø82,2Ø,AAA0,8
        Ø82,2Ø,2AA0,8ØØ2
BB 176Ø DATA AØ,A28,2ØA,AØ,AA0A,
        AA8,Ø,AA02
HA 177Ø DATA 28A0,Ø,AAØØ,AØ8Ø,Ø,
        AAØØ,8Ø82,Ø
OL 178Ø DATA AAØØ,8A,Ø,AAØØ,A8,Ø
        ,AAØØ,AØ
HA 179Ø DATA Ø,AØØØ,Ø,Ø,AØØØ,Ø,Ø
        ,AØØØ
BE 18ØØ DATA Ø,Ø,AØØØ,Ø,Ø,AØØØ,Ø
        ,Ø
PA 181Ø DATA AØØØ,Ø,Ø,AØØØ,Ø,Ø,A
        ØØA,Ø
LP 182Ø DATA Ø,Ø
EA 183Ø REM R1
QP 184Ø DATA 48,21,Ø,A,Ø,2Ø,8Ø22
        ,Ø
IC 185Ø DATA A8,8ØAA,Ø,8ØØA,2A,Ø
        ,AØØØ,A
BO 186Ø DATA Ø,AØØ,8ØAA,Ø,Ø,AØAA
        ,Ø,Ø
CF 187Ø DATA AAAA,Ø,Ø,82AA,AØ,Ø,
        8ØAA,8Ø2A
AE 188Ø DATA Ø,8ØAA,2,Ø,8ØAA,Ø,Ø
        ,8ØAA
KE 189Ø DATA Ø,Ø,8ØA2,Ø,2ØØ,8Ø82
        ,Ø,AØØ
MA 19ØØ DATA 8ØØ2,Ø,AØØ,8ØØ2,Ø,A
        Ø0,8ØØ2,Ø
JD 191Ø DATA AØØ,AØØØ,Ø,AAØØ,2ØØ
        Ø,Ø,Ø,AØØØ
LB 192Ø DATA Ø,Ø
FN 193Ø REM R2
CI 194Ø DATA 44,21,A8,28,Ø,A,8A,
        Ø
GA 195Ø DATA 82Ø2,AA,Ø,AØØØ,A8,Ø
        ,28ØØ,28
HI 196Ø DATA Ø,AØØ,AA,Ø,2ØØ,8ØAA
        ,Ø,2ØØ
MD 197Ø DATA AØAA,Ø,2ØØ,28AA,Ø,2
        ØØ,AAA,Ø
KH 198Ø DATA 2ØØ,2AA,8Ø,2ØØ,AA,A
        8,2ØØ,AA
KL 199Ø DATA 2Ø,2ØØ,8A,Ø,AØØ,A,Ø
        ,28ØØ
AA 2ØØØ DATA A,Ø,28ØØ,A,Ø,28ØØ,A
        ,Ø
NE 2Ø1Ø DATA A8Ø2,8ØØ2,Ø,Ø,8ØØØ,
        Ø,Ø,8Ø02
KA 2Ø2Ø DATA Ø,Ø
EH 2Ø3Ø REM R3
CC 2Ø4Ø DATA 42,21,Ø,AA0,Ø,2ØØ,8
        282,8Ø
LL 2Ø5Ø DATA A8Ø,AØØ8,AØ,28ØØ,AØ
        2A,AØ,28AØ,8ØØA
HC 2Ø6Ø DATA AØ,AA8,8202,8Ø,2ØA,
        AAAA,Ø,8Ø02
JF 2Ø7Ø DATA A8AA,Ø,AØØØ,AØ2A,Ø,
        28ØØ,AØ2A,Ø
BB 2Ø8Ø DATA AØØ,AØ2A,Ø,2ØØ,AØAA
        ,Ø,Ø,AØAA
CA 2Ø9Ø DATA Ø,Ø,AØØØ,Ø,Ø,AØØØ,Ø
        ,Ø
MA 21ØØ DATA AØØØ,Ø,Ø,AØØØ,Ø,Ø,A
        ØØØ,Ø
HH 211Ø DATA Ø,AØØØ,Ø,Ø,AØØØ,Ø,Ø
        ,AAØØ
KC 212Ø DATA Ø,Ø
HD 213Ø REM fig 12
MP 214Ø DATA 32,21,AA0,AA0,8A0,A
        2Ø,A28,28A0
BD 215Ø DATA 80A,A020,8202,8082,
        AA0Ø,A8,2A0Ø,A8
IO 216Ø DATA 2A0Ø,A8,2A0Ø,A8,2A0
        Ø,A8,2A0Ø,A8
FB 217Ø DATA 2A0Ø,A8,2A0Ø,A8,280
        Ø,28,AØØØ,A
FE 218Ø DATA 8Ø02,8Ø02,8Ø02,8Ø02
        ,8Ø02,8Ø02,8Ø02,8Ø02
MD 219Ø DATA 8Ø02,8Ø02,8Ø2A,A8Ø2
        ,Ø
IL 22ØØ REM fig 13
HB 221Ø DATA 32,18,Ø,2A,Ø,8Ø2A,Ø
        ,8Ø2A
HN 222Ø DATA Ø,8ØØA,Ø,Ø,Ø,AØØ2,8
        Ø02,A80A
CB 223Ø DATA AØØ2,AA2A,28ØØ,AAA0
        ,2Ø0,AA8Ø,2Ø,AAØØ
CP 224Ø DATA 2A28,AA0Ø,AA08,AA8Ø
        ,828A,AAAA,82,AA2A
AF 225Ø DATA AØ,AA0Ø,AAAA,AAAA,A
        A2A,A8AA,Ø
IM 226Ø REM block
PK 227Ø DATA 16,8,5050,5050,505,
        505,5050,5050
BO 228Ø DATA 505,505,Ø
BB 229Ø REM cross
QO 23ØØ DATA 16,8,550,1414,5005,
        5005,5005,5005
MA 231Ø DATA 1414,550,Ø
EA 232Ø REM ladder
DM 233Ø DATA 16,8,28Ø,28Ø,28Ø,28
        Ø,AAAA,28Ø
AA 234Ø DATA 28Ø,28Ø,Ø
DF 235Ø REM purple head
OL 236Ø DATA 16,8,A002,A80A,A80A
        ,A002,A82A,AAAA
JB 237Ø DATA AAAA,AAAA,Ø
NO 238Ø REM white head
KJ 239Ø DATA 16,8,F003,FC0F,FC0F
        ,F003,FC3F,FFFF
NG 24ØØ DATA FFFF,FFFF,Ø
KK 241Ø REM blue head
NO 242Ø DATA 16,8,5001,5405,5405
        ,5001,5415,5555
GH 243Ø DATA 5555,5555,Ø
```

Program 4: Amiga Tightrope

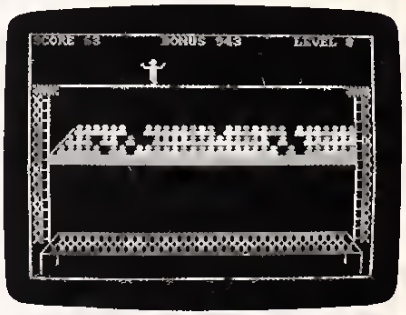
Version by Patrick Parrish, Programming Supervisor

Please refer to the typing instructions in this article before entering this listing.

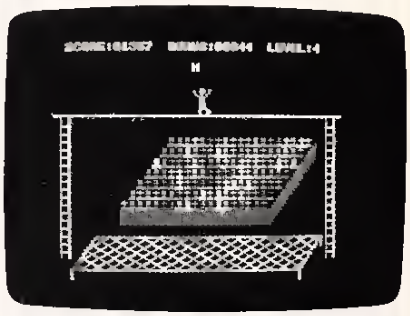
```
Ø GOSU8 setup:GOTO 7Ø←
1 PUT (X,Y),w3,PSET:RETURN←
2 PUT (X,Y),w2,PSET:RETURN←
3 PUT (X,Y),w1,PSET:RETURN←
4 PUT (X,Y),11,PSET:RETURN←
5 PUT (X,Y),12,PSET:RETURN←
6 PUT (X,Y),13,PSET:RETURN←
7 PUT (X,Y),r1,PSET:RETURN←
8 PUT (X,Y),r2,PSET:RETURN←
9 PUT (X,Y),r3,PSET:RETURN←
1Ø PUT (X,Y),w4,PSET:RETURN←
2Ø JV=Ø:a$=UCASE$(INKEY$)←
IF a$=CHR$(47) THEN ←
JV=1:AV=Ø:RETURN←
END IF←
IF a$=CHR$(9Ø) THEN←
JV=2:AV=Ø:RETURN ←
END IF←
21 AV=AV+1←
IF AV=b*4 THEN AV=Ø:RETURN←
22 GOTO 2Ø←
23 r1=INT(26*RND(1))+1←
X9=INT(X/8):r2=r1+64←
a=Ø:LOCATE 3,X9:PRINT CHR$(r2)←
24 a$=UCASE$(INKEY$):a=a+1←
IF a=b THEN←
M1=2←
GOSUB 177Ø:GOTO 3Ø←
END IF←
25 IF a$="" THEN 24←
26 IF a$=CHR$(r2) THEN←
M1=1←
GOSUB 174Ø:GOTO 3Ø←
END IF←
27 M1=2:GOSU8 177Ø←
3Ø LOCATE 3,X9:PRINT " "←
RETURN←
31 'T3=3*INT(TIMER-T2)←
T=1ØØØ-T3:IF T<Ø THEN T=Ø←
32 RETURN←
7Ø CLS:FOR X=Ø TO 36 STEP 36←
FOR J=Ø TO 2←
PUT (X*8+J*8,4Ø),s14:NEXT J,X←
LINE (24,4Ø)-(287,4Ø),3←
75 FOR J=Ø TO 38 STEP 38←
FOR I=1 TO 15←
PUT (J*8,I*8+4Ø),s15←
PUT (J*8+8+(J=38)*16,I*8+4Ø),s16
←
NEXT I,J←
8Ø FOR r=1 TO 3:FOR c=6-r TO 36←
X=RND(1):ROW=64+8*r:COL=c*8←
IF X<.3 THEN←
PUT(COL,ROW),s17:GOTO 1ØØ←
END IF←
85 IF X>=.3 AND X<.6 THEN←
PUT(COL,ROW),s18:GOTO 1ØØ←
END IF←
9Ø IF X>=.6 AND X<.9 THEN←
PUT(COL,ROW),s19:GOTO 1ØØ←
END IF←
95 LOCATE ROW/8,COL/8:PRINT " "←
1ØØ NEXT c,r←
1Ø5 LINE (39,75)-(17,96)←
LINE (38,75)-(16,96)←
FOR I=96 TO 96+7←
LINE (16,I)-(8*37-1,I):NEXT I←
```

```
107 LINE (23,20*8-1)-(36*8,20*8-
1)
LINE (7,22*8)-(38*8,22*8)
LINE (23,20*8)-(7,22*8)
LINE (36*8,20*8-1)-(38*8,22*8)
108 FOR r=0 TO 1
FOR c=3-r TO 35+r
PUT (c*8,(r+20)*8),s15
NEXT c,r
109 LINE (7,22*8)-(7,24*8)
LINE (38*8,22*8)-(38*8,24*8)
LINE (23,22*8)-(23,23*8)
LINE (36*8,22*8)-(36*8,23*8)
160 Y=19:PX=7:PZ=10:c=0
440 T2=TIMER
500 IF w=2 AND d=8 THEN
b=100
END IF
IF w=2 AND d=9 THEN
b=65
END IF

510 P=0:GOSUB 3000
FOR X=288 TO 4 STEP -3
520 c=c+d+1:GOSUB 31
550 LOCATE 1,1
PRINT "SCORE"c"     "TAB(16)"BONU
S"T" ";
PRINT TAB(31)"LEVEL"d
600 f=0:IF X<268 AND X>24 THEN G
OSUB 780
605 IF f=1 THEN 1080
630 P=P+1:IF P=5 THEN P=1
640 ON P GOSUB 1,2,3,2
655 JV=0:a$=UCASE$(INKEY$):IF a$
=" " THEN GOSUB 1510
IF a$=CHR$(47) THEN JV=1:GOSUB 8
10
660 IF a$=CHR$(90) THEN JV=2:GOS
UB 950
665 IF f=1 THEN 1080
670 NEXT X:X=X+3
GOSUB 3000:GOSUB 10
710 c=c+T:T=0
720 IF d<9 THEN b=b-2:d=d+1
750 IF w=2 AND d<9 THEN b=b-25
760 IF d=9 THEN PX=6:PZ=9
770 GOTO 440
780 r=INT(9*RND(1))+1
IF r>2 THEN RETURN
800 IF r=1 THEN 950
810 P=4:PUT (X,Y),11,PSET
820 IF w=1 THEN GOSUB 20
IF w=2 THEN GOSUB 23
840 IF M1=1 THEN JV=2
IF M1=2 THEN JV=1
860 IF JV=2 THEN
P=P-1
ON P GOSUB ,,3,4,5,6
IF P<4 THEN RETURN ELSE 820
END IF
880 P=P+1:ON P GOSUB ,,,4,5,6
890 GOSUB 31
910 LOCATE 1,21:PRINT T
920 IF P>3 AND P<PX THEN 820
930 f=1:RETURN
950 P=7:PUT (X,Y),r1,PSET
960 IF w=1 THEN GOSUB 20
IF w=2 THEN GOSUB 23
980 IF M1=1 THEN JV=1
IF M1=2 THEN JV=2
990 IF JV=1 THEN P=P-1
1010 GOSUB 31
1030 LOCATE 1,21:PRINT T
1040 IF P<7 THEN P=3:RETURN
1050 IF JV=1 THEN
ON P GOSUB ,,,,,,7,8,9
GOTO 960
END IF
1060 P=P+1
ON P GOSUB ,,,,,,7,8,9
1070 IF P>6 AND P<PZ THEN 960
1075 f=1:RETURN
1080 P=10
PUT (X,Y),w4,PSET
'FALLING MAN
1090 Z2=23
FOR Z=Z2 TO 150 STEP 6
IF Z>Z2 THEN PUT (X,Z-6),s12
1100 PUT (X,Z),s12
SOUND (Z+15)*2,.08
1110 NEXT Z
PUT (X,Z-6),s12:Y=Y+50
1120 FOR Z=150 TO Y STEP -6
IF Z<150 THEN PUT (X,Z+6),s13
1125 PUT (X,Z),s13
SOUND (Z+15)*2,1:NEXT Z
PUT (X,Z+6),s13
1130 FOR Z=Y TO 150 STEP 6
IF Z>Y THEN PUT (X,Z-6),s13
1135 PUT (X,Z),s13
SOUND (Z+15)*2,1:NEXT Z
PUT (X,Z-6),s13:Y=Y+30
IF Y<150 THEN 1120
1160 PUT (X,Z-6),s13:LOCATE 3,6
PRINT "Press <RETURN> to play ag
ain"
LOCATE 4,7
PRINT "Press <SPACE BAR> for men
u"
1170 a$=INKEY$
IF a$<>" " AND a$<>CHR$(13) THEN
1170
1180 c=0:M1=0:AV=0
IF a$=CHR$(13) THEN 70
GOSUB again:GOTO 70
1510 T4=INT(TIMER)
1520 a$=INKEY$
IF a$="" THEN 1520
1530 T5=INT(TIMER):T2=T2+T5-T4
RETURN
1740 SOUND 440,1:RETURN
1770 SOUND 2300,1:RETURN
3000 FOR DE=1 TO 400:NEXT:RETURN

setup:
DEFINT L,r,s,w
SCREEN 1,320,200,2,1
' OPEN WINDOW 3 WITH NO GADGETS
OR
' title BAR
WINDOW 1,"",(0,0)-(311,25),16,1
WINDOW 3,"",(0,0)-(311,185),16,1


WINDOW OUTPUT 3
PALETTE 0,0,0,0
PALETTE 1,.5,1,1
PALETTE 2,1,0,0
PALETTE 3,1,1,.1
WIDTH 40
CLS
DIM voice%(8),w4%(200)
GET (0,0)-(25,20),w4%
RESTORE VOICEDATA
FOR J=0 TO 8
READ voice%(J)
NEXT
' Speech will be synchronous
VOICEDATA:
DATA 110,0,170,0,22200,64,10,1,0


talk$="Welcome to Tightrope"
LOCATE 12,11
PRINT talk$
GOSUB talk
L= 87 :DIM w1%(L)
FOR I=0 TO L:READ a$:w1%(I)=VAL(
"&h"+a$):NEXT
shapedata:
DATA 18,15,2,30,0,58,0,F8
DATA 0,4070,800,C030,1800,2078,2
000,1FFF
DATA C000,F8,0,F8,0,F8,0,F8
DATA 0,F8,0,F8,0,D8,0,D8
DATA 0,D8,0,78,0,38,0,1C
DATA 0,18,0,78,0,30,0,78
DATA 0,F8,0,4070,800,C030,1800,2
078
DATA 2000,1FFF,C000,F8,0,F8,0,F8
DATA 0,F8,0,F8,0,F8,0,D8
DATA 0,198,0,318,0,318,0,318
DATA 0,18C,0,186,0,30C,0,0
L= 87 :DIM w2%(L)
FOR I=0 TO L:READ a$:w2%(I)=VAL(
"&h"+a$):NEXT
DATA 18,15,2,30,0,58,0,F8
DATA 0,4070,800,C030,1800,2078,2
000,1FFF
DATA C000,F8,0,F8,0,F8,0,F8
DATA 0,F8,0,F8,0,D8,0,198
DATA 0,318,0,318,0,318,0,18C
DATA 0,186,0,30C,0,30,0,78
DATA 0,F8,0,4070,800,C030,1800,2
078
DATA 2000,1FFF,C000,F8,0,F8,0,F8
DATA 0,F8,0,F8,0,F8,0,D8
DATA 0,198,0,318,0,318,0,318
DATA 0,18C,0,186,0,30C,0,0
L= 87 :DIM w3%(L)
FOR I=0 TO L:READ a$:w3%(I)=VAL(
"&h"+a$):NEXT
DATA 18,15,2,30,0,58,0,F8
DATA 0,4070,800,C030,1800,2078,2
000,1FFF
DATA C000,F8,0,F8,0,F8,0,F8
DATA 0,F8,0,F8,0,D8,0,18C
DATA 0,306,0,603,0,601,8000,600
DATA C000,600,C000,1E01,8000,30,
0,78
DATA 0,F8,0,4070,800,C030,1800,2
078
DATA 2000,1FFF,C000,F8,0,F8,0,F8
DATA 0,F8,0,F8,0,F8,0,D8
DATA 0,18C,0,306,0,603,0,601
DATA 8000,600,C000,600,C000,1E01
,8000,0
L= 87 :DIM r1%(L)
FOR I=0 TO L:READ a$:r1%(I)=VAL(
"&h"+a$):NEXT
DATA 18,15,2,30,0,58,1000,F8
DATA 1800,70,7000,31,C000,7F,0,1
FC
DATA 0,7F8,0,1CF8,0,F0F8,0,40F8
DATA 0,F8,0,F8,0,D8,0,198
DATA 0,318,0,30C,0,306,0,186
DATA 0,18C,0,300,0,30,0,78
DATA 1000,F8,1800,70,7000,31,C00
0,7F
DATA 0,1FC,0,7F8,0,1CF8,0,F0F8
DATA 0,40F8,0,F8,0,F8,0,D8
DATA 0,198,0,318,0,30C,0,306
DATA 0,186,0,18C,0,300,0,0
L= 87 :DIM r2%(L)
FOR I=0 TO L:READ a$:r2%(I)=VAL(
"&h"+a$):NEXT
DATA 18,15,2,30,3800,58,6000,F8
DATA C000,71,8000,33,0,7E,0,1FC
DATA 0,3F8,0,6F8,0,CF8,0,18F8
DATA 0,30F8,0,E0F8,0,D8,0,198
DATA 0,318,0,30C,0,307,C000,180
DATA C000,180,4000,300,0,30,3800
,78
DATA 6000,F8,C000,71,8000,33,0,7
E
DATA 0,1FC,0,3F8,0,6F8,0,CF8
DATA 0,18F8,0,30F8,0,E0F8,0,D8
DATA 0,198,0,318,0,30C,0,307
DATA C000,180,C000,180,4000,300,
0,0
L= 87 :DIM r3%(L)
FOR I=0 TO L:READ a$:r3%(I)=VAL(
"&h"+a$):NEXT
DATA 15,15,2,1860,798,3330,330,6
598
DATA 47C2,CF98,478C,C718,C7B0,63
31,C7C0,3FE3
DATA 400,1FC6,400,F8C,400,F98,79
8,FB0
DATA 330,FE0,7C2,FC0,78C,C00,7B0
,C00
```

```
DATA 7C0,C00,400,C00,400,C00,400
,C00<
DATA 400,C00,400,3C00,0,1860,0,3
330<
DATA 0,6798,4000,CF98,4000,C718,
C000,6331<
DATA C000,3FE3,0,1FC6,0,F8C,0,F9
8<
DATA 0,FB0,0,FE0,0,FC0,0,C00<
DATA 0,C00,0,C00,0,C00,0,C00<
DATA 0,C00,0,C00,0,3C00,0,0<
L= 87 :DIM 11%(L)<
FOR I=0 TO L:READ a$:11%(I)=VAL(
"&h"+a$):NEXT<
DATA 18,15,2,0,98,30,30,4058<
DATA C2,E0F8,8C,3870,B0,E30,C0,3
F8<
DATA 0,FC,0,FF,0,F9,C098,F8<
DATA 7830,F8,10C2,F8,8C,D8,80,19
8<
DATA C0,318,0,318,0,318,0,30C<
DATA 0,F04,0,C,0,0,0,30<
DATA 0,4078,0,E0F8,0,3870,0,E30<
DATA 0,3F8,0,FC,0,FF,0,F9<
DATA C000,F8,7800,F8,1000,F8,0,D
8<
DATA 0,198,0,318,0,318,0,318<
DATA 0,30C,0,F04,0,C,0,0<
L= 87 :DIM 12%(L)<
FOR I=0 TO L:READ a$:12%(I)=VAL(
"&h"+a$):NEXT<
DATA 16,15,2,0,98,E060,30,30B0<
DATA C2,19F0,8C,CE0,B0,660,C0,3F
0<
DATA 0,1F8,0,1FC,0,1F6,98,1F3<
DATA 30,1F1,80C2,1F0,E08C,1B0,40
B0,330<
DATA C0,630,0,630,0,630,0,618<
DATA 0,1E08,0,18,0,0,0,E060<
DATA 0,30F0,0,19F0,0,CE0,0,660<
DATA 0,3F0,0,1F8,0,1FC,0,1F6<
DATA 0,1F3,0,1F1,8000,1F0,E000,1
80<
DATA 4000,330,0,630,0,630,0,630<
DATA 0,618,0,1E08,0,18,0,0<
L= 87 :DIM 13%(L)<
FOR I=0 TO L:READ a$:13%(I)=VAL(
"&h"+a$):NEXT<
DATA 14,15,2,C3,98,199,8030,832C
<
DATA C0C2,867C,C08C,C638,C0B0,E3
19,80C0,31FF<
DATA 0,18FE,0,C7C,0,67C,98,37C<
DATA 30,1FC,C2,FC,8C,C,B0,C<
DATA C0,C,0,C,0,C,0,C<
DATA 0,C,0,F,0,C3,0,199<
DATA 8000,833C,C000,867C,C000,C6
38,C000,E319<
DATA 8000,31FF,0,18FE,0,C7C,0,67
C<
DATA 0,37C,0,1FC,0,FC,0,C<
DATA 0,C,0,C,0,C,0,C<
DATA 0,C,0,C,0,F,0,0<
L= 87 :DIM 14%(L)<
FOR I=0 TO L:READ a$:14%(I)=VAL
("&h"+a$):NEXT<
DATA 13,15,2,C3C3,98,C243,30,63C
6<
DATA C2,324C,8C,1998,B0,FF0,C0,7
E0<
DATA 0,7E0,0,7E0,0,7E0,98,7E0<
DATA 30,7E0,C2,7E0,8C,660,B0,C30
<
DATA C0,1818,0,1818,0,1818,0,181
8<
DATA 0,1818,0,781E,0,C3C3,0,C3C3
<
DATA 0,63C6,0,324C,0,1998,0,FF0<
DATA 0,7E0,0,7E0,0,7E0,0,7E0<
DATA 0,7E0,0,7E0,0,7E0,0,660<
DATA 0,C30,0,1818,0,1818,0,1818<
DATA 0,1818,0,1818,0,781E,0,0<
L= 39 :DIM 15%(L)<
FOR I=0 TO L:READ a$:15%(I)=VAL
("&h"+a$):NEXT<
DATA 10,12,2,70,78,78,38,0<
DATA 1C,183E,1C7F,6CF,18F,400F,6
70F,2F8F<
DATA B9FF,907F,C00F,FFFF,7FFE,70
,78,78<
DATA 38,0,1C,183E,1C7F,6CF,18F,4
00F<
DATA 670F,2F8F,89FF,907F,C00F,FF
FF,7FFE,1818<
L= 19 :DIM 18%(L)<
FOR I=0 TO L:READ a$:18%(I)=VAL
("&h"+a$):NEXT<
DATA 8,8,2,1C00,3E00,3E00,1C00,7
E00<
DATA FF00,FF00,FF00,0,0,0,0,0<
DATA 80,80,80,0<
L= 19 :DIM 17%(L)<
FOR I=0 TO L:READ a$:17%(I)=VAL
("&h"+a$):NEXT<
DATA 8,8,2,0,0,0,0,0<
DATA 0,0,0,1C00,3E00,3E00,1C00,7
E00<
DATA FF80,FF80,FF80,0<
L= 19 :DIM 19%(L)<
FOR I=0 TO L:READ a$:19%(I)=VAL
("&h"+a$):NEXT<
DATA 8,8,2,1C00,3E00,3E00,1C00,7
E00<
DATA FF00,FF00,FF00,1C00,3E00,3E
00,1C00,7E00<
DATA FF80,FF80,FF80,0<
L= 19 :DIM 15%(L)<
FOR I=0 TO L:READ a$:15%(I)=VAL
("&h"+a$):NEXT<
DATA 8,8,2,C300,6600,3C00,3C00,3
C00<
DATA 3C00,6600,C300,0,0,0,0,0<
DATA 0,0,0,0<
L= 19 :DIM 16%(L)<
FOR I=0 TO L:READ a$:16%(I)=VAL
("&h"+a$):NEXT<
DATA 8,8,2,8100,8100,8100,8100,F
F00<
DATA 8100,8100,8100,8100,8100,81
00,8100,FF00<
DATA 8100,8100,8100,0<
L= 19 :DIM 14%(L)<
FOR I=0 TO L:READ a$:14%(I)=VAL
("&h"+a$):NEXT<
DATA 8,8,2,70,78,78,38,0<
DATA 1C,3E,7F,CCCF,CC8F,330F,330
F,CC8F<
DATA CCFF,337F,330F,FFFF<
again:<
RANDOMIZE TIMER<
<
CLS<
GOSUB title<
RETURN<
<
announce:<
talk$=c$<
<
talk:<
IF talkflag=0 THEN SAY TRANSLATE
$(talk$),voice%<
RETURN<
<
title:<
talk$="press 1 for game, 2 for t
yping"<
GOSUB talk<
WINDOW 4,"       Press 1 or 2",(6
5,70)-(250,110),16,1<
PRINT:PRINT "          1- Game"<
PRINT:PRINT "          2- Typing"<
<
grabkey:<
a$=INKEY$:IF a$="" THEN grabkey<
w=VAL(a$)<
IF w<1 OR w>2 THEN grabkey<
talk$="Press 0 through 9 to choo
ss difficulty level."<
GOSUB talk<
WINDOW 4,"Press 0-9 for difficul
ty",(65,70)-(255,110),16,1<
PRINT:PRINT:PRINT "    Enter lev
el (0-9)"<
<
grabkey1:<
a$=INKEY$:IF a$="" OR (a$<"0" OR
a$>"9") THEN grabkey1<
b=VAL(a$)<
d=b:b=10-b:IF w=1 THEN b=b*2<
IF w=2 THEN b=b*40<
WINDOW CLOSE 4<
temp$="typing"<
IF w=1 THEN temp$="game" <
talk$=temp$+"      "+"level"+STR
$(d)<
GOSUB talk<
RETURN<
<
gstout:<
WINDOW CLOSE 3<
BCREEN CLOSE 1<
WINDOW 1,"Tightrope",,31,-1<
WINDOW OUTPUT 1<
END<
```



Amiga "Tightrope" requires 512K of memory and uses keyboard controls.



This version of "Tightrope" runs on all Apple II computers using either a joystick or game paddles.

Program 5: Apple Tightrope

Version by Tim Victor, Editorial Programmer

```
25 10 HOME : VTAB 6: HTAB 8: PRI
      NT "READING DATA, PLEASE W
      AIT": GOSUB 770
E4 20 TEXT : HOME : VTAB 12: HTA
      B 15: PRINT "(1) GAME": HT
      AB 15: PRINT "(2) TYPING":
       GET A$:GM = VAL (A$): IF
      GM < 1 OR GM > 2 THEN 20
97 30 HOME : VTAB 12: HTAB 10: P
      RINT "DIFFICULTY(0-9) ";:
      GET L$: IF L$ < "0" OR L$
      > "9" THEN 30
52 40 GOSUB 590
CE 50 CALL 32771
58 60 LV = VAL (L$):SC = 0
C9 70 BO = 1000: GOSUB 520:I = 2
      5B
```

```
C8 B0 FOR S = 1 TO 3: CALL 32768
   ,B,I - S * 2,32: FOR J = 1
   TO 200: NEXT : NEXT
7C 90 I = I - 6: IF I > 228 THEN
   B0
EE 100 OF = 225 / (LV + 1.5):C =
   0
2E 110 FOR J = 1 TO 1000: NEXT
E2 120 IF RND (1) > .08 * (10 -
   LV) THEN GOSUB 320
B9 130 IF C = 1 THEN 280
FA 140 IF GM = 2 THEN 170
A6 150 IF POL (0) < 90 THEN A =
   4: GOSUB 330: GOTO 130
19 160 IF POL (0) > 156 THEN A =
    8: GOSUB 330: GOTO 130
1B 170 FOR S = 1 TO 3: CALL 3276
   8,S,I - S * 2,32
23 180 BD = BO - 3: IF BD < 0 TH
   EN BO = 0
15 190 SC = SC + LV + 1: GOSUB 5
   20
7B 200 FOR J = 1 TO 150: NEXT
BB 210 NEXT :I = I - 6: IF I > 1
   8 THEN 120
F1 220 LV = LV + 1: IF LV > 9 TH
   EN LV = 9
92 230 SC = SC + BO: GOSUB 520:
   FOR J = 1 TO 400: NEXT
16 240 FOR S = 1 TO 3: CALL 3276
   8,S,I - S * 2,32
56 250 FOR J = 1 TO 300: NEXT :
   NEXT
3F 260 I = I - 6: IF I > 0 THEN
   240
3B 270 GOTO 70
CD 280 VTAB 2: HTAB 7: PRINT "PR
   ESS RETURN TO PLAY AGAIN"
   : HTAB 8: PRINT "PRESS SP
   ACE BAR FOR MENU"
BC 290 POKE 49168,0: GET A$: IF
   A$ = CHR$ (32) THEN 20
BD 300 IF A$ = CHR$ (13) THEN VT
   AB 2: PRINT SPC( 80): GOT
   O 60
8A 310 HOME : TEXT : END
BD 320 A = ( RND (1) > .5) * 4 +
   4
73 330 CALL 32768,A,I,32
9A 340 IF GM = 1 THEN 400
74 350 C$ = CHR$ (65 + 26 * RND
   (1)): VTAB 3: HTAB INT (I
   / 7) + 1: PRINT C$
75 360 POKE 49168,0:CT = DF
95 370 K = PEEK (49152): IF K <
   128 THEN CT = CT - 1: IF
   CT > 0 THEN 370
77 380 IF K > 127 AND K - 128 =
   ASC (C$) THEN 500
48 390 PRINT CHR$ (7): GOTO 440
AD 400 CT = DF / 6
DB 410 IF POL (0) > 89 AND PDL (
   0) < 157 THEN CT = CT - 1
   : IF CT > 0 THEN 410
39 420 IF PDL (0) < 90 AND A > 7
    OR PDL (0) > 156 AND A <
   8 THEN 500
74 430 FOR J = 1 TO CT: NEXT
6D 440 A = A + 1:BD = BD - 3: IF
    A = 8 THEN A = 12
34 450 IF A < 12 THEN 330
9E 460 FOR T = 0 TO 40:Y = 142 -
    110 * EXP ( - T / 10) *
   ABS ( COS (T / 2))
19 470 CALL 32768,12,I,Y
33 480 FOR CT = 1 TO 12: NEXT :
   NEXT
A4 490 C = 1: VTAB 3: HTAB INT (
   I / 7) + 1: PRINT " ": RE
   TURN
C5 500 A = A - 1: IF A < > 3 AND
    A < > 7 THEN 330
7B 510 VTAB 3: HTAB INT (I / 7)
   + 1: PRINT " ":C = 0: RET
   URN
DB 520 A = FRE (0): VTAB 1: HTAB
    3: PRINT "SCORE:";: HTAB
   16: PRINT "BONUS:";: HTA
   B 29: PRINT "LEVEL:";
AF 530 HTAB 9:NL = 5:NN = SC: GO
   SUB 560
73 540 HTAB 22:NL = 5:NN = BO: G
   OSUB 560
63 550 HTAB 35:NL = 1:NN = LV
F2 560 NS$ = RIGHT$ ( STR$ (NN),
   NL)
11 570 IF LEN (NS$) < NL THEN NS
   $ = "0" + NS$: GOTO 570
74 580 PRINT NS$;: RETURN
AB 590 HGR2 : HCOLOR= 7
4F 600 HCOLOR= 6: FOR X = 1 TO 1
   2: HPLOT 62,126 + X TO 19
   B,126 + X TO 240,76 + X:
   NEXT
26 610 VTAB 10: FOR V = 1 TO 7:
   HTAB 17 - V: FOR H = 1 TO
    19: PRINT CHR$ (60 + INT
   (4 * RND (1)));: NEXT :
   PRINT : NEXT
34 620 HCOLOR= 7: FOR X = 160 TO
    60 STEP - 6: HPLOT 10,X
   TO 17,X: HPLOT 250,X TO 2
   57,X: NEXT
3E 630 FOR X = 0 TO 240 STEP 240
   : FOR Y = 0 TO 7 BTEP 7:
   HPLOT 10 + X + Y,164 TO 1
   0 + X + Y,56: NEXT : NEXT
B1 640 HPLDT 4,56 TO 263,56
B5 650 HCOLOR= 5: HPLDT 40,148 T
   O 245,148 TO 225,173 TO 2
   0,173 TO 40,148
E7 660 FOR X = - 12 TO 223 STEP
   12:BX = X:SY = 173:EX = X
    + 50:EY = 148
94 670 IF SX < 20 THEN SX = 30 -
    2 * X / 3:SY = 173 - SX
   / 2 + X / 2
37 680 IF EX > 245 THEN EX = 376
    - 2 * X / 3:EY = 173 - E
   X / 2 + X / 2
6E 690 HPLDT SX,SY TO EX,EY: NEX
   T
D5 700 FOR X = 20 TO 295 STEP 14
   :SX = X:SY = 173:EX = X -
   50:EY = 148
34 710 IF EX < 40 THEN EX = 15 +
    2 * X / 7:EY = 173 + EX
   / 2 - X / 2
AA 720 IF SX > 225 THEN SX = 161
    + 2 * X / 7:SY = 173 + S
   X / 2 - X / 2
63 730 HPLDT SX,SY TO EX,EY: NEX
   T
9C 740 HPLDT 19,173 TO 19,180: H
   PLDT 225,173 TO 225,180:
   HPLOT 245,148 TO 245,155
21 750 RETURN
FC 760 PRINT "DATA ERRDR": END
4E 770 FOR A = 768 TO A + 87: RE
   AD D: POKE A,D: NEXT
94 780 READ D: IF D < > - 1 THEN
    760
BB 790 FOR A = 32768 TO A + 502:
    READ O: POKE A,D: NEXT
B5 800 READ D: IF D < > - 1 THEN
    760
7C 810 FOR A = 33620 TO A + 863:
    READ D: POKE A,D: NEXT
B9 820 READ D: IF D < > - 1 THEN
    760
7A 830 FOR A = 36096 TO A + 7: P
   OKE A,128: NEXT
7E 840 FOR A = 36200 TO A + 367:
    READ D: POKE A,0: NEXT
8F 850 READ D: IF D < > - 1 THEN
    760
63 860 IF PEEK (190 * 256) = 76
   THEN PRINT CHR$ (4);"PR#A
   768": GOTO 880
C7 870 POKE 54,0: POKE 55,3: CAL
   L 1002
58 880 POKE 6,0: POKE 7,141: POK
   E 230,64
2A 890 RETURN
79 900 DATA 216,120,133,69,134,7
   0,132,71,166,7,10
9D 910 DATA 10,176,4,16,62,48,4,
   16,1,232,232
AE 920 DATA 10,134,27,24,101,6,1
   33,26,144,2,230
FC 930 DATA 27,165,40,133,8,165,
   41,41,3,5,230
9B 940 DATA 133,9,162,8,160,0,17
   7,26,36,50,48
C8 950 DATA 2,73,127,164,36,145,
   8,230,26,208,2
74 960 DATA 230,27,165,9,24,105,
   4,133,9,202,208
E7 970 DATA 226,165,69,166,70,16
   4,71,88,76,240,253
C3 980 DATA -1
DB 990 DATA 76,6,128,76,71,128,1
   73,9,130,240,3
69 1000 DATA 32,77,128,32,141,12
   8,176,57,32,140,129
1F 1010 DATA 176,52,32,227,129,1
   76,47,173,249,129,133
33 1020 DATA 252,141,7,130,173,2
   50,129,133,253,141,8
59 1030 DATA 130,173,255,129,141
   ,4,130,173,2,130,141
1C 1040 DATA 5,130,173,3,130,141
   ,6,130,169,255,141
55 1050 DATA 9,130,76,102,128,16
   9,0,141,9,130,96
CE 1060 DATA 173,7,130,133,252,1
   73,8,130,133,253,173
B3 1070 DATA 4,130,141,255,129,1
   73,5,130,141,2,130
51 1080 DATA 173,6,130,169,2,141
   ,251,129,169,24,141
8E 1090 DATA 252,129,32,36,129,3
   2,205,128,32,22,129
6B 1100 DATA 238,255,129,165,252
   ,24,105,3,133,252,144
EB 1110 DATA 2,230,253,206,252,1
   29,208,228,96,169,12
BB 1120 DATA 141,249,129,169,131
   ,141,250,129,32,236,129
B7 1130 DATA 201,21,144,1,96,141
   ,247,129,169,0,141
FE 1140 DATA 248,129,160,3,32,17
   4,128,160,3,173,247
91 1150 DATA 129,10,46,248,129,1
   36,208,249,141,247,129
A7 1160 DATA 24,109,249,129,141,
   249,129,173,250,129,109
B2 1170 DATA 248,129,141,250,129
   ,24,96,172,251,129,200
F1 1180 DATA 140,253,129,169,0,1
   53,12,130,136,177,252
6A 1190 DATA 153,12,130,136,16,2
   48,173,12,130,9,127
3C 1200 DATA 141,254,129,172,3,1
   30,240,21,162,0,14
BB 1210 DATA 12,130,189,12,130,1
   0,62,13,130,232,236
FB 1220 DATA 253,129,208,243,136
   ,208,235,172,253,129,185
37 1230 DATA 12,130,9,128,45,254
   ,129,153,12,130,136
66 1240 DATA 16,242,96,172,253,1
   29,185,12,130,81,254
C4 1250 DATA 145,254,136,16,246,
   96,173,255,129,41,63
EB 1260 DATA 168,185,76,129,5,23
   0,133,255,173,255,129
A8 1270 DATA 41,8,240,2,169,128,
   24,44,255,129,112
1B 1280 DATA 4,16,4,105,40,105,4
   0,109,2,130,133
71 1290 DATA 254,96,0,4,8,12,16,
   20,24,28,0
99 1300 DATA 4,8,12,16,20,24,28,
   1,5,9,13
```

```
EA 1310 DATA 17,21,25,29,1,5,9,1
       3,17,21,25
35 1320 DATA 29,2,6,10,14,18,22,
       26,30,2,6
01 1330 DATA 10,14,18,22,26,30,3
       ,7,11,15,19
6A 1340 DATA 23,27,31,3,7,11,15,
       19,23,27,31
9F 1350 DATA 32,236,129,140,0,13
       0,141,158,129,169,0
1F 1360 DATA 141,1,130,24,160,4,
       105,216,162,3,106
6A 1370 DATA 110,1,130,24,202,20
       8,248,136,208,241,141
4B 1380 DATA 2,130,173,1,130,42,
       42,42,42,41,7
27 1390 DATA 201,7,208,5,238,2,1
       30,169,0,141,3
DF 1400 DATA 130,173,0,130,240,2
       3,24,173,3,130,105
EE 1410 DATA 4,201,7,144,2,169,0
       ,141,3,130,173
54 1420 DATA 2,130,105,36,141,2,
       130,201,40,96,32
89 1430 DATA 236,129,141,255,129
       ,201,192,96,32,177,0
E2 1440 DATA 32,5,225,165,161,16
       4,160,96
23 1450 DATA -1
58 1460 DATA 0,0,0,0,0,0,0,28,0,
       0,60
59 1470 DATA 0,0,62,0,0,28,0,0,1
       2,0,6
43 1480 DATA 0,48,84,42,21,0,42,
       0,0,42,0
CD 1490 DATA 0,42,0,0,42,0,0,42,
       0,0,42
00 1500 DATA 0,0,62,0,0,62,0,0,5
       1,0,64
FA 1510 DATA 97,0,64,97,0,64,65,
       1,64,65,1
71 1520 DATA 64,1,3,96,64,1,0,0,
       0,0,0
7C 1530 DATA 0,0,28,0,0,60,0,0,6
       2,0,0
E7 1540 DATA 28,0,0,12,0,6,0,48,
       84,42,21
0F 1550 DATA 0,42,0,0,42,0,0,42,
       0,0,42
F7 1560 DATA 0,0,42,0,0,42,0,0,6
       2,0,0
03 1570 DATA 62,0,0,54,0,0,54,0,
       0,99,0
5D 1580 DATA 0,67,1,0,6,3,0,70,1
       ,0,3
0C 1590 DATA 0,0,0,0,0,0,0,0,0,28,
       0,0
92 1600 DATA 60,0,0,62,0,0,28,0,
       0,12,0
3A 1610 DATA 6,0,48,84,42,21,0,4
       2,0,0,42
E1 1620 DATA 0,0,42,0,0,42,0,0,4
       2,0,0
9F 1630 DATA 42,0,0,62,0,0,62,0,
       0,54,0
00 1640 DATA 0,30,0,0,30,0,0,60,
       0,0,108
0E 1650 DATA 0,0,56,0,0,12,0,0,0
       ,0,0
D7 1660 DATA 0,0,0,28,0,0,60,0,0
       ,62,0
6B 1670 DATA 0,28,0,0,12,48,0,0,
       21,6,42
CA 1680 DATA 0,84,42,0,0,42,0,0,
       42,0,0
D5 1690 DATA 42,0,0,42,0,0,42,0,
       0,62,0
50 1700 DATA 0,62,0,0,54,0,0,54,
       0,0,99
FC 1710 DATA 0,0,67,1,0,6,3,0,70
       ,1,0
35 1720 DATA 3,0,0,0,0,0,0,0,0,0,2
       8,0
03 1730 DATA 0,60,0,0,62,24,0,28
       ,8,0,12
99 1740 DATA 4,0,0,1,0,42,0,64,4
       2,0,16
66 1750 DATA 42,0,8,42,0,12,42,0
       ,0,42,0
50 1760 DATA 0,42,0,0,62,0,0,62,
       0,0,54
A2 1770 DATA 0,0,102,0,0,67,1,0,
       3,6,0
CF 1780 DATA 6,3,0,6,0,0,3,0,0,0
       ,0
FD 1790 DATA 0,0,0,0,28,0,0,60,4
       8,0,62
F8 1800 DATA 16,0,28,4,0,12,4,0,
       0,1,2
64 1810 DATA 42,0,86,42,0,0,42,0
       ,0,42,0
0A 1820 DATA 0,42,0,0,42,0,0,42,
       0,0,62
4F 1830 DATA 0,0,62,0,0,102,0,0,
       70,1,0
0D 1840 DATA 3,3,0,3,6,0,6,3,0,6
       ,0
0E 1850 DATA 0,3,0,0,0,0,0,0,0,0
       ,0
DC 1860 DATA 0,0,0,0,0,28,0,0,60
       ,12,12
00 1870 DATA 62,4,8,28,4,16,12,1
       ,64,0,1
D1 1880 DATA 0,42,0,0,42,0,0,42,
       0,0,42
02 1890 DATA 0,0,42,0,0,42,0,0,4
       2,0,0
D6 1900 DATA 62,0,0,126,0,0,102,
       1,0,3,7
D6 1910 DATA 0,3,12,0,6,7,0,3,0,
       0,0
73 1920 DATA 0,0,0,0,0,28,0,0,60
       ,0,0
C9 1930 DATA 62,0,6,28,0,4,12,0,
       80,0,0
DB 1940 DATA 0,42,0,0,42,5,0,42,
       48,0,42
F3 1950 DATA 0,0,42,0,0,42,0,0,4
       2,0,0
D4 1960 DATA 62,0,0,62,0,0,54,0,
       0,124,0
61 1970 DATA 0,124,1,0,12,3,0,10
       8,1,0,16
21 1980 DATA 0,0,12,0,0,0,0,0,0,
       0,0
86 1990 DATA 28,0,0,60,0,12,62,0
       ,8,28,0
09 2000 DATA 16,12,0,64,0,0,0,42
       ,0,0,42
68 2010 DATA 1,0,42,4,0,42,48,0,
       42,0,0
2F 2020 DATA 42,0,0,42,0,0,62,0,
       0,62,0
0D 2030 DATA 0,60,0,0,124,0,0,76
       ,7,0,12
52 2040 DATA 6,0,12,3,0,16,0,0,1
       2,0,0
A7 2050 DATA 0,0,0,0,0,0,28,0,6,
       60,0
9C 2060 DATA 4,62,0,16,28,0,16,1
       2,0,64,0
30 2070 DATA 0,0,42,32,0,42,53,0
       ,42,0,0
86 2080 DATA 42,0,0,42,0,0,42,0,
       0,42,0
DD 2090 DATA 0,62,0,0,60,0,0,120
       ,0,0,24
0E 2100 DATA 7,0,24,12,0,24,6,0,
       24,0,0
0C 2110 DATA 24,0,0,12,0,0,0,0,0
       ,0,0
6F 2120 DATA 0,0,0,0,0,0,0,0,28,0,
       24,60
0D 2130 DATA 0,16,62,24,16,28,8,
       64,12,4,64
68 2140 DATA 0,1,0,42,0,0,42,0,0
       ,42,0
00 2150 DATA 0,42,0,0,42,0,0,42,
       0,0,42
9C 2160 DATA 0,0,62,0,0,60,0,0,1
       20,1,0
06 2170 DATA 24,6,0,24,3,0,48,0,
       0,28,0
01 2180 DATA 0,0,0,0,0,0,0,0,0,0
       ,0
72 2190 DATA 0,0,0,0,0,0,0,0,28,
       0,24
6D 2200 DATA 60,12,16,62,4,16,28
       ,4,64,12,1
35 2210 DATA 64,0,1,0,42,0,0,42,
       0,4,46
09 2220 DATA 0,4,46,0,12,46,0,24
       ,46,0,48
D2 2230 DATA 58,0,96,63,0,64,127
       ,0,0,119,0
2D 2240 DATA 0,0,0,0,0,0
20 2250 DATA -1
A5 2260 DATA 128,128,128,128,190
       ,128,128,128,128,128,128
02 2270 DATA 190,128,190,128,128
       ,128,188,230,176,152,128
5E 2280 DATA 152,128,128,188,230
       ,246,238,230,188,128,128
49 2290 DATA 152,156,152,152,152
       ,188,128,128,188,230,176
99 2300 DATA 140,230,254,128,128
       ,188,230,176,224,230,188
D0 2310 DATA 128,128,176,184,180
       ,254,176,176,128,128,254
1A 2320 DATA 134,190,224,230,188
       ,128,128,188,134,190,230
5A 2330 DATA 230,188,128,128,254
       ,224,176,152,140,140,128
D6 2340 DATA 128,188,230,188,230
       ,230,188,128,128,188,230
23 2350 DATA 230,252,176,152,128
       ,128,128,152,152,128,152
0E 2360 DATA 152,128,128,190,190
       ,190,190,190,190,128,128
7F 2370 DATA 148,148,148,128,148
       ,213,213,0,20,20,20
59 2380 DATA 0,20,85,85,128,136,
       170,136,128,136,170
1D 2390 DATA 170,0,8,42,8,0,8,42
       ,42,128,128
34 2400 DATA 152,188,188,152,128
       ,128,128,252,230,230,254
E0 2410 DATA 230,230,128,128,190
       ,230,230,190,230,254,128
3E 2420 DATA 128,188,230,134,134
       ,230,190,128,128,190,230
50 2430 DATA 230,230,230,190,128
       ,128,254,134,134,190,134
2A 2440 DATA 254,128,128,254,134
       ,134,190,134,134,128,128
69 2450 DATA 188,230,134,246,230
       ,190,128,128,230,230,230
2A 2460 DATA 254,230,230,128,128
       ,152,152,152,152,152,152
46 2470 DATA 128,128,224,224,224
       ,224,230,188,128,128,230
1F 2480 DATA 230,182,158,230,230
       ,128,128,134,134,134,134
00 2490 DATA 134,254,128,128,194
       ,230,254,230,230,230,128
2D 2500 DATA 128,190,230,230,230
       ,230,230,128,128,188,230
2E 2510 DATA 230,230,230,188,128
       ,128,190,230,230,190,134
8E 2520 DATA 134,128,128,188,230
       ,230,230,182,236,128,128
8E 2530 DATA 190,230,230,190,230
       ,230,128,128,188,230,140
CA 2540 DATA 176,230,190,128,128
       ,254,152,152,152,152,152
00 2550 DATA 128,128,230,230,230
       ,230,230,190,128,128,230
04 2560 DATA 230,230,230,230,152
       ,128,128,230,230,230,254
92 2570 DATA 230,194,128,128,230
       ,230,164,152,164,230,128
4F 2580 DATA 128,230,230,230,188
       ,152,152,128,128,254,176
57 2590 DATA 152,140,134,254,128
14 2600 DATA -1
```

©

SOFTBALL STATISTICS FOR ATARI ST

Roger Felton

"Softball Statistics" makes it easy to keep track of all the individual and team results for your favorite team. You can enter data for each player's times at bat, hits, runs, and so on. The program automatically computes batting averages, stores cumulative results on disk as the season progresses, generates formatted printouts with sorted rankings for all players, and more. The program was originally written for the eight-bit Atari and adapted for several other computers in the July 1985 issue of COMPUTE!*. This new version runs in medium- or high-resolution modes on any Atari ST with the TOS operating in ROM. An 80-column printer is optional but recommended.*

What's the worst position on a softball team? Catchers have to squat in an uncomfortable stance for an hour or more and duck hazardous foul balls. Pitchers have to duel with mighty sluggers and dodge powerful line drives. First basemen have to stretch their bodies like rubber bands to nab wayward throws from their teammates while keeping at least one toe on the base bag. And outfielders have to scoop up bouncing grounders with the knowledge that no one is backing them up except the outfield fence.

But as demanding as all these positions are, there's another that could be worse—that of team statistician. Keeping track of your teammates' performance is often a laborious, thankless job. Sometimes the statistician is a reserve player or friend of the team who doesn't even get to play. Caged in the dugout, the statistician is supposed to document every hit, run, and walk, and boost team morale by contributing lively chatter. After the game, the statistician has to spend hours punching numbers into a calculator to figure out everyone else's batting average.

"Softball Statistics" makes that job much easier. After each game, the program prompts you to enter vital stats for each player. Then it automatically calculates the batting averages and prints sorted rankings on the screen or printer. It can also print sorted rankings for hits, runs, and runs batted in. These game statistics can then be merged with data for all previous games, and updated season results can be sorted by category and printed. Finally, the program lets you store the cumulative statistics on disk.

If you're a fan of professional or Little League baseball, you can use Softball Statistics to follow the fortunes of your favorite team. And with modifications, it could be adapted to a wide variety of sports.

Preparing The Program

Be extra careful when typing Softball Statistics because a mistyped line could yield inaccurate results even if the program runs without errors. Save a copy on disk for safekeeping before running it the first time.

Before using the program, you have to prepare it by entering your team's roster. Softball Statistics can handle a team with up to 20 players and stores this information in DATA statements as part of the program itself. If you're keeping stats for more than one team, you'll have to keep a separate copy of the program for each team.

The DATA statements for player information begin at line 2300. The statements must conform to a predefined format: a two-digit jersey number followed by a space, then the player's first or last name.

Precede one-digit jersey numbers with a zero, such as 08 for 8. Names can be any length, but only the first seven characters appear on the printouts. Each entry is separated by a comma. Example:

```
2300 DATA 23 LEE,17 JACKSON,33
     JOHNSTON,10 LONGSTREET,04
     PICKETT
```

(In the output, JOHNSTON and LONGSTREET would appear as JOHNSTO and LONGSTR.)

The programs are listed with dummy entries in the DATA statements, such as 44 Jim and 10 PLAYERX. Substitute your own team members for these entries. If your team has fewer than 20 players, leave the remaining dummy entries in the DATA statements but substitute the name PLAYERX; the program must have 20 entries to function, and it ignores the PLAYERX entries.

Finally, put your own team's name in the TM$ string statement at line 190. Softball Statistics is now ready to run.

Important note: You should avoid tinkering with the player name DATA statements once you've started using the program. Otherwise, there will be problems when it attempts to compute cumulative season totals. If you drop a player from the roster and replace him with another player, the new player's totals will contain the old player's results as well. To drop a player, substitute a PLAYERX dummy entry at that position in the DATA statement. Of course, this means the dropped player's results will no longer be included in the team totals for the season. If you wish to retain a dropped player's results in the team totals, leave the player's name in the DATA statement and enter 999 in response to all input prompts for that player's stats following subsequent games (see below).

Compiling Statistics

Once the roster is entered, you can run the program. It begins by asking for statistics for individual games. The first prompt asks:

Who did you play?

Respond with the opposing team's name—such as Ham's Diner—and press RETURN. The next prompt reads:

Figure 1: Printout of Team Game Stats

ROSTER IS SORTED BY BATTING AVERAGE

#	PLAYER	AB	RUNS	HITS	RBI	2B	3B	HR	BB	AVG
09	MARTY	6	2	5	3	2	1	1	0	0.833
03	JOHN	5	2	4	2	2	0	1	1	0.800
55	MIKE	4	1	3	1	1	0	1	0	0.750
44	JIM	5	4	3	1	2	0	0	0	0.600
0B	KEN	4	1	2	1	1	1	0	0	0.500
0B	BOB	6	3	3	2	2	0	0	2	0.500
22	PETE	5	1	2	2	0	0	0	0	0.400
07	BILL	5	1	2	0	1	0	0	0	0.400
06	BARRY	6	2	2	0	1	0	0	3	0.333
TOTALS		46	17	26	12	12	2	3	6	0.565

Enter your score and their score (separated by a comma):

For instance, if your team lost by a score of 9 to 5, you'd type 5,9 and press RETURN.

The program now begins asking for individual player statistics. If the first player name on your roster is Kevin, the program prints

Kevin's statistics for this game:

and then prompts you, one by one, to enter the number of times at bat, runs scored, hits, runs batted in (RBIs), doubles, triples, home runs, and walks. At each prompt, type the appropriate number and press RETURN. After the last prompt, the program asks:

Is everything OK (Y/N)?

If you made any mistakes while entering the current player's stats, press N. You'll be given a chance to reenter the numbers.

When all the player's statistics are correct, press Y at the prompt. The program continues to the next player on the roster and repeats the cycle.

If a certain player missed a game, type 999 at the first prompt. This automatically enters zeros for all his stats and skips to the next player. In fact, entering 999 at any prompt inputs zeros for all of a player's remaining game stats.

Individual Printouts

After you type the last statistic for the last player, the program prints the message WORKING while it sorts all the data. (The WORKING message appears at other points in the program during sorts, since the sort routine is written in BASIC and is not particularly fast.) In a few moments, the program says:

Do you want a printout of the game's stats (Y/N)?

Type Y for yes or N for no. If

Figure 2: Printout of Slugging Stats

HITS SORT:

#	PLAYER	HITS
09	MARTY	5
03	JOHN	4
55	MIKE	3
44	JIM	3
0B	BOB	3
06	BARRY	2
0B	KEN	2
22	PETE	2
07	BILL	2
TOTAL HITS		26

RBIS SORT:

#	PLAYER	RBIS
09	MARTY	3
03	JOHN	2
22	PETE	2
0B	BOB	2
44	JIM	1
55	MIKE	1
0B	KEN	1
07	BILL	0
06	BARRY	0
TOTAL RBIS		12

RUNS SORT:

#	PLAYER	RUNS
44	JIM	4
0B	BOB	3
03	JOHN	2
06	BARRY	2
09	MARTY	2
55	MIKE	1
0B	KEN	1
22	PETE	1
07	BILL	1
TOTAL RUNS		17

you press N, the program asks if you want to input data for another game. If you press Y, it asks:

To screen or printer (S/P)?

Type S or P. Softball Statistics then prints the individual stats for all team members for that game, sorted in descending order by batting averages (see Figure 1). To pause the printout, press the left mouse button. You can resume after pausing by pressing the space bar.

Next, the program asks:

Do you want a sorted printout of hits, RBIs, and run leaders (Y/N)?

Again, type Y for yes or N for no. If you type N, the program asks if you want to input stats for another game. If you answer Y, it asks again if you want the output directed to the screen or printer, and then prints sorted rankings for the various slugging categories for that game (see Figure 2). As before, you can stop the output by pressing the left mouse button and restart it by pressing the space bar.

Finally, the program asks:

Do you want to input stats from another game (Y/N)?

Usually you type N at this prompt unless you're entering results of more than one game. If you type Y, the program repeats the entire process described above.

Season Totals

Softball Statistics makes it easy for you to tabulate running totals for the entire season by storing game results on disk. After you've entered and viewed the stats for the most recent game, the program asks:

Would you like to merge in data for the year (Y/N)?

The first time you run Softball Statistics, of course, you won't have any previous data on disk, so you'd answer N, skipping to the next prompt. During subsequent runs, you'd answer Y to merge in data for the year. The program then requests a filename for the disk data file and merges these existing stats with the results you've entered for the latest game or games.

Season totals are then computed automatically, and the program asks:

Do you want a printout of the year's stats (Y/N)?

Figure 3: Printout of Season Totals

```
STATISTICS FOR THE YEAR:
RECORD FOR THE YEAR:  WINS:2 LOSSES:1

ROSTER IS SORTED BY BATTING AVERAGE

#  PLAYER    AB   RUNS   HITS   RBI    2B    3B    HR    BB     AVG
03 JOHN      16     10     11    11     5     4     2     3    0.688
06 BARRY     18     12     11     8     4     1     4     5    0.611
07 BILL      17     10     10     7     3     3     3     2    0.588
55 MIKE      18     10     10    10     5     3     1     4    0.556
44 JIM       18      9      9     7     5     2     1     2    0.500
08 BOB       17     12      8     7     4     1     2     1    0.471
09 MARTY     17     10      8    10     4     2     3     4    0.471
22 PETE      17      7      6     4     3     1     1     3    0.353
08 KEN       17      6      6     7     3     1     2     4    0.353

TOTALS      155     86     79    71    36    18    19    28    0.510
```

If you answer Y, the program asks if you want output directed to the screen or printer, and then prints season totals for all players. This printout includes the team's win-loss record and sorts players in descending order by batting averages (see Figure 3).

Afterward, the program asks if you want sorted printouts for hits, RBIs, and runs—again, based on season totals (these charts resemble those in Figure 2). Finally, the program gives you the opportunity to save the updated data file on disk until the next game.

If you typed N after the previous prompt, the program asks:

Do you want to save the data (Y/N)?

If you answer Y, the program asks for a filename for the updated data file, saves the file, and then ends.

Softball Computing

If you're interested in programming, you can learn a lot by studying Softball Statistics because it's written in straight BASIC with no machine language. In fact, the input and output routines beginning at lines 2350 and 2470 are general enough to be adapted to your own programs.

You don't have to be a programmer, though, to appreciate Softball Statistics. If you're a softball statistician, no longer do you have the worst position on the team. Maybe it's the shortstop....

Softball Statistics For Atari ST

Version By George Miller, Assistant Technical Editor

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
10    TITLE$=" Softball Statist
      ics "+CHR$(0)
20    LP$=SPACE$(2)+"# PLAYER"+
      SPACE$(4)+"AB"+SPACE$(3)
30    LP$=LP$+"RUNS"+SPACE$(2)+
      "HITS"+SPACE$(3)+"RBI"+SPA
      CE$(3)
40    LP$=LP$+"2B"+SPACE$(4)+"3
      B"+SPACE$(4)+"HR"+SPACE$(4
      )+"BB"+SPACE$(4)+"AVG"
50    GOSUB CLEARIT
60    IF PEEK(SYSTAB+0) <> 4 TH
      EN 140
70    PRINT " 'SOFTBALL STATIST
      ICS'"
80    PRINT " REQUIRES A MEDIUM
       OR HI RESOLUTION"
90    PRINT " SCREEN ":PRINT
100   PRINT " PLEASE USE THE CO
      NTROL PANEL"
110   PRINT " TO RESET RESOLUTI
      ON BEFORE"
120   PRINT " CONTINUING "
130   END
140   GOSUB CLEARIT:GOSUB TITLE
      BAR
150   D5=5
160   D6=2
170   PL=20
180   DIM B(9),CC(20),IN(21),ST
      (8),RT(20,8),TT(20,8),F$(8
      ),NA$(20),R$(21)
190   TM$="Sundogs"
200   C$="0000"
210   FOR I=1 TO 8
220   READ F$(I)
230   NEXT I
240   FOR J=1 TO PL
250   READ NA$(J)
260   NA$(J)=MID$(NA$(J),1,10)
270   NEXT J
280   FOR J=1 TO PL
290   R$(J)=MID$(NA$(J),1,LEN(N
      A$(J)))+SPACE$(10-LEN(NA$(
      J)))
300   FOR I=1 TO 8
310   TT(J,I)=0
320   ST(I)=0
```

```
330   NEXT I
340   NEXT J
350   GOSUB CLEARIT:GOTOXY 5,10
      :PRINT "Do you want to:":P
      RINT
360   PRINT SPACE$(20);"1) Ente
      r new statistics "
365   PRINT SPACE$(20);"2) Revi
      ew disk file"
370   A = INP(2)
380   IF A = ASC("1") THEN 410
390   IF A = ASC("2") THEN 3530

400   GOTO 370
410   GOSUB CLEARIT:PRINT "GAME
       STATISTICS"
420   PRINT:PRINT "Who did you
      play"
430   INPUT OT$
440   PRINT:PRINT "Enter your s
      cora and their score (sepa
      rated by a comma)"
450   INPUT YS,TS
460   W=W+ABS(YS>TS)
470   L=L+ABS(TS>YS)
480   FOR J=1 TO PL
490   IF MID$(NA$(J),4,7)<>"PLA
      YERX" THEN 520
500   R$(J)=R$(J)+"000000000000
      00000000000000000000 000"

510   GOTO 730
520   GOSUB CLEARIT
530   PRINT MID$(NA$(J),4,LEN(N
      A$(J)));"'s statistics for
       this game:"
540   FOR I=1 TO 8
550   B(I)=0
560   PRINT F$(I);TAB(14);
570   INPUT B(I)
580   IF LEN(STR$(B(I)))>=05 TH
      EN 550
590   IF B(I)<>999 THEN 640
600   FOR K=I TO 8
610   B(K)=0
620   NEXT K
630   I=8
640   NEXT I
650   PRINT:PRINT"Is avarything
       OK (Y/N) ?"
660   A$ = CHR$(INP(2))
670   IF A$ = "N" OR A$ = "n" T
      HEN 520
680   GOSUB BUILOR
690   FOR I=1 TO 8
700   RT(J,I)=RT(J,I)+B(I)
710   TT(J,I)=TT(J,I)+B(I)
720   NEXT I
730   NEXT J
740   GOSUB WORKING
750   MM=0
760   FOR I=1 TO 8
770   FOR J=1 TO PL
780   ST(I)=ST(I)+TT(J,I)
790   NEXT J
800   B(I)=ST(I)
810   NEXT I
820   R$(J)=""
830   GOSUB BUILOR
840   TT$=R$(J)
850   GOSUB AVERAGE:GOSUB CLEAR
      IT
860   PRINT "Oo you want to inp
      ut statistics from another
       game (Y/N)?"
870   GOSUB GETKEY
880   IF A$ = "Y" OR A$ = "y" T
      HEN 280
890   GOSUB CLEARIT
900   PRINT "Would you like to
      merge in data for the year
       (Y/N)?"
910   GOSUB GETKEY
920   IF A$ = "N" OR A$ = "n" T
      HEN 960
930   GOSUB CHECKFILE
940   W=SW+W
950   L=SL+L
960   GOSUB WORKING
970   FOR J=1 TO PL
980   FOR I=1 TO 8
990   IF A$="N" OR A$="n" OR MI
      D$(NA$(J),4,7)="PLAYERX" T
      HEN 1040
1000  B(I)=VAL(MID$(R$(J),11+(I
      -1)*4,4))
1010  B(I)=RT(J,I)+B(I)
1020  RT(J,I)=B(I)
1030  GOTO 1050
1040  B(I)=RT(J,I)
1050  ST(I)=0
1060  NEXT I
1070  R$(J)=MID$(R$(J),1,10)
1080  GOSUB BUILDR
1090  NEXT J
1100  MM=1
1110  FOR I=1 TO 8
1120  FOR J=1 TO PL
1130  ST(I)=ST(I)+RT(J,I)
1140  NEXT J
1150  B(I)=ST(I)
1160  NEXT I
1170  R$(J)=""
1180  GOSUB BUILDR
1190  TT$=R$(J)
1200  GOSUB CLEARIT
1210  PRINT "Do you want a prin
      tout of the year's statist
      ics (Y/N)?"
1220  GOSUB GETKEY
1230  IF A$ = "N" OR A$ = "n" T
      HEN 1260
1240  GOSUB WORKING
1250  GOSUB AVERAGE:GOSUB CLEAR
      IT
1260  PRINT "Do you want to SAV
      E tha data (Y/N)?"
1270  GOSUB GETKEY
1280  IF A$ = "Y" OR A$ = "y" T
      HEN 1300
1290  END
1300  GOTO WRITEFILE
1310  '
1320  SHELL:
1330  FOR J=1 TO PL
1340  IN(J)=J
1350  CC(J)=VAL(MID$(R$(J),BB,E
      ))
1360  NEXT J
1370  FOR J=PL-1 TO 1 STEP -1
1380  FOR I=1 TO J
1390  IF CC(IN(I))>CC(IN(I+1))
      THEN 1430
1400  TE=IN(I)
1410  IN(I)=IN(I+1)
1420  IN(I+1)=TE
1430  NEXT I
1440  NEXT J
1450  RETURN
1460  '
1470  BUILOR:
1480  IF B(1)=0 THEN 1510
1490  IF B(3)=0 THEN 1510
1500  GOTO 1540
1510  B(9)=0
1520  AV$="0 000"
1530  GOTO 1550
1540  B(9)=INT(B(3)/B(1)*1000+
      5)/1000+ 0001
1550  FOR I=1 TO 8
1560  B$=STR$(B(I))
1570  B$=MID$(C$,1,05-LEN(B$))+
      MIO$(B$,06,LEN(B$))
1580  R$(J)=R$(J)+B$
1590  NEXT I
1600  IF B(9)=0 THEN 1660
1610  AV$=STR$(B(9))
1620  IF MIO$(AV$,1,1)<>" " THE
      N 1640
1630  AV$=MIO$(AV$,2,6)
1640  IF MIO$(AV$,1,1)<>"." THE
      N 1660
1650  AV$="0"+AV$
1660  R$(J)=R$(J)+MIO$(AV$,1,5)
1670  RETURN
1680  '
1690  AVERAGE:
1700  BB=43
1710  E=5
1720  GOSUB SHELL
1730  IF MM=1 THEN 1770
1740  GOSUB CLEARIT
1750  PRINT "Do you want a prin
      tout of the game's statist
      ics (Y/N)?"
1760  GOSUB GETKEY
1770  IF A$ = "N" OR A$ = "n" T
      HEN 1810
1780  GOSUB PRINTOPT
1790  IF OE = 1 THEN GOSUB SCRE
      ENPRNT:GOTO 1810
1800  IF DE = 2 THEN GOTO LINEP
      RNT
1810  RETURN
1820  '
1830  WORKING:
1840  PRINT
1850  PRINT " WORKING  ."
1860  RETURN
1870  '
1880  PRINT
1890  PRINT "Oo you want sorted
       printouts of hits, RBI's,
       and run laadars (Y/N)?"
1900  GOSUB GETKEY
1910  IF A$ = "N" OR A$ = "n" T
      HEN 1940
1920  GOSUB PRINTOPT
1930  GOTO 1950
1940  RETURN
1950  GOSUB WORKING
1960  BB=19
1970  E=4
1980  GOSUB SHELL
1990  I=3
2000  IF DE = 1 THEN GOSUB TOSC
      REEN ELSE GOSUB TOLINEPTR
2010  BB=23
2020  GOSUB SHELL
2030  I=4
2040  IF DE = 1 THEN GOSUB TOSC
      REEN ELSE GOSUB TOLINEPTR
2050  BB=15
2060  GOSUB SHELL
2070  I=2
2080  IF OE = 1 THEN GOSUB TOSC
      REEN ELSE GOSUB TOLINEPTR
2090  RETURN
2100  '
2110  GETKEY:
2120  A$ = CHR$(INP(2))
2130  IF A$ = "N" OR A$ = "n" O
      R A$ = "Y" OR A$ = "y" THE
      N RETURN ELSE 2120
2140  RETURN
2150  '
2160  PRINTOPT:
2170  PRINT
2180  PRINT "To scraan or print
      er (S/P)?"
2190  A$ = CHR$(INP(2))
2200  IF A$ = "S" OR A$ = "s" T
      HEN OE = 1:GOTO 2220
2210  IF A$ = "P" OR A$ = "p" T
      HEN OE = 2 ELSE 2190
2220  RETURN
2230  '
2240  CLEARIT:
2250  CLEARW 2:FULLW 2:GOTOXY 0
      ,0
2260  RETURN
2270  '
2280  OATA Timas at Bat,Runs,Hi
      ts,RBI's,Ooubles,Triples,H
      ome Runs,Walks
2290  REM LIST PLAYERS BY NUMBE
      R S NAME
2300  OATA 01 Kavin,02 Tom,03 P
      atrick,04 Eddia,05 Gragg
2310  OATA 06 Georga,07 Oavid H
      ,08 Oavid F.,09 Selby,10
```

```
      Mark
2320  DATA 11 Neal,12 Byron,13
      Paul,14 John,15 Leon
2330  DATA 16 David K,17 Mike,1
      8 PLAYERX,19 PLAYERX,20 PL
      AYERX
2340  '
2350  REM INPUT ROUTINE
2360  CHECKFILE:
2370  ON ERROR GOTO 2600
2380  GOSUB CLEARIT
2390  PRINT"Name for data file"
      ;:INPUT FF$
2400  OPEN "I",#1,FF$
2410  INPUT #1,SW,SL
2420  FOR J=1 TO PL
2430  INPUT #1,R$(J)
2440  R$(J)=MIO$(NA$(J),1,LEN(N
      A$(J)))+SPACE$(10-LEN(NA$(
      J)))+R$(J)
2450  NEXT J:CLOSE #1:RETURN
2460  '
2470  WRITEFILE:
2480  GOSUB CLEARIT:
2490  PRINT"Name of data file t
      o write";:INPUT FF$
2500  OPEN "O",#1,FF$
2510  PRINT#1,W
2520  PRINT#1,L
2530  FOR J = 1 TO PL
2540  PRINT #1, MIO$(R$(J),11,3
      2)
2550  NEXT J
2560  CLOSE #1
2570  ENO
2580  '
2590  CHECKERROR:
2600  IF ERR = 53 THEN 2620
2610  PRINT "Error Number ";ERR
      ;" at line ";ERL:ENO
2620  PRINT "File not found on
      disk drive specified "
2630  CLOSE 1
2640  RESUME 2390
2650  '
2660  SCREENPRNT:
2670  GOSUB CLEARIT:PRINT:IF MM
      =1 THEN T$="THE YEAR":GOTO
       2890
2680  T$="THIS GAME"
2690  PRINT "STATISTICS FOR "T$
      ":":IF MM=1 THEN GOTO 2710

2700  PRINT TM$" VS "OT$"    Sc
      ore:"YS"-"TS:GOTO 2720
2710  PRINT "RECORD FOR THE YEA
      R:  Wins-"W" Losses:"L
2720  PRINT :PRINT "Roster is s
      orted by batting average":
      PRINT
2730  PRINT LP$
2740  FOR J=1 TO PL:GOSUB PAUSE

2750  IF MIO$(R$(IN(J)),4,7)="P
      LAYERX" THEN 2830
2760  PRINT SPACE$(1);MIO$(R$(I
      N(J)),1,10);SPACE$(1);
2770  FOR I= 1 TO 8:Q=0:FOR K=0
       TO 3
2780  IF MID$(R$(IN(J)),11+(I-1
      )*4+K,1) <> "0" THEN D=1
2790  IF MIO$(R$(IN(J)),11+(I-1
      )*4+K,1)="0" AND Q=0 AND K
      =3 THEN PRINT "0";:GOTO 28
      20
2800  IF MID$(R$(IN(J)),11+(I-1
      )*4+K,1)="0" AND O=0 THEN
      PRINT " ";:GOTO 2820
2810  PRINT MIO$(R$(IN(J)),11+(
      I-1)*4+K,1);
2820  NEXT K:PRINT SPACE$(2);:N
      EXT I:PRINT SPACE$(1);MIO$
      (R$(IN(J)),43,5)
2830  NEXT J:PRINT :PRINT " TOT
      ALS";SPACE$(5);
2840  FOR I=1 TO 8
2850  O=0:FOR K=1 TO 4:IF MIO$(
      TT$,(I-1)*4+K,1) <> "0" TH
      EN Q=1
2860  IF MID$(TT$,(I-1)*4+K,1)=
      "0" AND Q=0 AND K=4 THEN P
      RINT "0";:GOTO 2890
2870  IF MID$(TT$,(I-1)*4+K,1)=
      "0" AND Q=0 THEN PRINT SPA
      CE$(1);:GOTO 2890
2880  PRINT MID$(TT$,(I-1)*4+K,
      1);
2890  NEXT K:PRINT SPACE$(2);:N
      EXT I:PRINT SPACE$(1);MIO$
      (TT$,33,5)
2900  PRINT :GOTO 1880
2910  '
2920  TOSCREEN:
2930  PRINT :T=0:PRINT :PRINT F
      $(I)" SORT:":PRINT
2940  PRINT "#  PLAYER";space$(
      6);F$(I):FOR J=1 TO PL:GDS
      UB PAUSE
2950  IF MID$(R$(IN(J)),4,7)="P
      LAYERX" THEN 3020
2960  PRINT MIO$(R$(IN(J)),1,10
      );SPACE$(4);
2970  Q=0:FOR K=0 TO 3:IF MID$(
      R$(IN(J)),BB+K,1) <> "0" T
      HEN O=1
2980  IF MID$(R$(IN(J)),BB+K,1)
      ="0" AND Q=0 AND K=3 THEN
       PRINT "0":GOTO 3010
2990  IF MIO$(R$(IN(J)),BB+K,1)
      ="0" AND D=0 THEN PRINT SP
      ACE$(1);:GDTD 3010
3000  PRINT MIO$(R$(IN(J)),BB+K
      ,1);:IF K=3 THEN PRINT
3010  NEXT K:T=T+VAL(MIO$(R$(IN
      (J)),BB,E))
3020  NEXT J:PRINT :PRINT "TDTA
      L ";F$(I);SPACE$(5);T
3030  PRINT : RETURN
3040  '
3050  LINEPRNT:
3060  LPRINT:IF MM=1THEN T$="TH
      E YEAR":GOTO 3080
3070  T$="THIS GAME"
3080  LPRINT "STATISTICS FOR "T
      $":":IF MM=1 THEN GOTO 310
      0
3090  LPRINT TM$" V9 "OT$"    S
      CORE:"YS"-"TS:GDTO 3110
3100  LPRINT "Record for the ye
      ar:  Wins:"W" Losses:"L
3110  LPRINT :LPRINT "Roster is
       sorted by Batting Average
      ":LPRINT
3120  LPRINT LP$
3130  FOR J=1 TO PL:GOSUB PAUSE

3140  IF MIO$(R$(IN(J)),4,7)="P
      LAYERX" THEN 3220
3150  LPRINT SPACE$(1);MIO$(R$(
      IN(J)),1,10);SPACE$(1);
3160  FOR I=,1 TO 8:O=0:FOR K=0
       TO 3
3170  IF MIO$(R$(IN(J)),11+(I-1
      )*4+K,1) <> "0" THEN O=1
3180  IF MID$(R$(IN(J)),11+(I-1
      )*4+K,1)="0" AND Q=0 AND K
      =3 THEN LPRINT "0";:GOTO 3
      210
3190  IF MIO$(R$(IN(J)),11+(I-1
      )*4+K,1)="0" AND Q=0 THEN
      LPRINT " ";:GOTO 3210
3200  LPRINT MID$(R$(IN(J)),11+
      (I-1)*4+K,1);
3210  NEXT K:LPRINT SPACE$(2);:
      NEXT I:LPRINT SPACE$(1);MI
      O$(R$(IN(J)),43,5)
3220  NEXT J:LPRINT:LPRINT " TO
      TALS"+SPACE$(5);
3230  FOR I=1 TO 8
3240  Q=0:FOR K=1 TO 4:IF MID$(
      TT$,(I-1)*4+K,1) <> "0" TH
      EN O=1
3250  IF MIO$(TT$,(I-1)*4+K,1)=
      "0" AND Q=0 AND K=4 THEN L
      PRINT "0";:GOTO 3280
3260  IF MID$(TT$,(I-1)*4+K,1)=
      "0" AND Q=0 THEN LPRINT SP
      ACE$(1);:GOTO 3280
3270  LPRINT MID$(TT$,(I-1)*4+K
      ,1);
3280  NEXT K:LPRINT SPACE$(2);:
      NEXT I:LPRINT SPACE$(1);MI
      D$(TT$,33,6)
3290  LPRINT:GOTO 1880
3300  '
3310  TOLINEPTR:
3320  LPRINT :T=0:LPRINT :LPRIN
      T F$(I)" SORT:":LPRINT
3330  LPRINT "#"+SPACE$(2)+"PLA
      YER"+SPACE$(6);F$(I):FOR J
      =1 TO PL:GOSUB PAU9E
3340  IF MIO$(R$(IN(J)),4,7)="P
      LAYERX" THEN 3410
3350  LPRINT MID$(R$(IN(J)),1,1
      0)SPACE$(4);
3360  Q=0:FOR K=0 TO 3:IF MID$(
      R$(IN(J)),BB+K,1) <> "0" T
      HEN Q=1
3370  IF MID$(R$(IN(J)),BB+K,1)
      ="0" AND Q=0 ANO K=3 THEN
       LPRINT "0":GOTO 3400
3380  IF MIO$(R$(IN(J)),BB+K,1)
      ="0" ANO Q=0 THEN LPRINT S
      PACE$(1);:GOTD 3400
3390  LPRINT MIO$(R$(IN(J)),BB+
      K,1);:IF K=3 THEN LPRINT
3400  NEXT K:T=T+VAL(MIO$(R$(IN
      (J)),BB,E))
3410  NEXT J:LPRINT :LPRINT "TO
      TAL ";F$(I);SPACE$(5);T
3420  LPRINT:RETURN
3430  '
3440  PAUSE:
3450  IF PEEK(&HFFFC02) > 0 THE
      N 3450 ELSE RETURN
3460  '
3470  TITLEBAR:
3480  A# = GB : GINTIN = PEEK(A
      #+8)
3490  POKE GINTIN+0,PEEK(SYSTAB
      +8) : POKE GINTIN+2,2
3500  S# = GINTIN+4 : TITLE$ =
      TITLE$ + CHR$(0)
3510  POKE S#,VARPTR(TITLE$) :
      GEMSYS(105)
3520  RETURN
3530  REVIEW:
3540  GOSUB CHECKFILE
3550  W=SW+W
3560  L=SL+L
3570  GOSUB WORKING
3580  FOR J=1 TO PL
3590  FOR I=1 TO 8
3600  IF A$="N" OR A$="n" OR MI
      D$(NA$(J),4,7)="PLAYERX" T
      HEN 3620
3610  B(I)=VAL(MID$(R$(J),11+(I
      -1)*4,4))
3620  B(I)=RT(J,I)+B(I)
3630  RT(J,I)=B(I)
3640  GOTO 3660
3650  B(I)=RT(J,I)
3660  ST(I)=0
3670  NEXT I
3680  R$(J)=MIO$(R$(J),1,10)
3690  GOSUB BUILDR
3700  NEXT J
3710  MM=1
3720  FOR I=1 TO 8
3730  FOR J=1 TO PL
3740  ST(I)=ST(I)+RT(J,I)
3760  NEXT J
3760  B(I)=ST(I)
3770  NEXT I
3780  R$(J)=""
3790  GOSUB BUILOR
3800  TT$=R$(J)
3810  GOSUB CLEARIT
3820  GOSUB WORKING
3830  GOSUB AVERAGE:GOSUB CLEAR
      IT
3840  ENO                    ©
```

Toshiba P321 Printer

Tim Victor, Editorial Programmer

Requirements: Any compatible computer with the appropriate interface.

A few years ago it was easy to spend a lot of money for a computer printer and still not get top-of-the-line quality. Unless you were satisfied with a 40-column thermal printer, you generally had to lay out several hundred dollars just to get a relatively crude dot-matrix printer, and a good daisy wheel printer cost over $1,000.

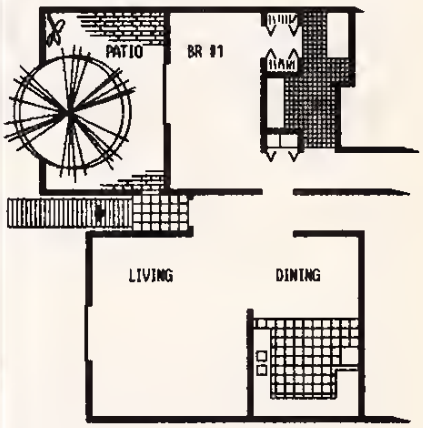
Today, many good dot-matrix and daisy wheel printers are available for a couple of hundred dollars. But both technologies have their particular strengths, and which one you ultimately choose should depend on the applications you have in mind. Daisy wheel printers necessarily have limited graphics capabilities, though they offer letter-quality type. Inexpensive dot-matrix printers can produce decent graphics, but are restricted in print quality by their nine-pin printheads, which generally don't produce letter-quality type or crisp graphics (though some have very respectable near-letter-quality modes).

Dot-matrix printers that use a 24-pin printhead are capable of far superior graphics and text, but have in the past been relatively expensive. Toshiba has introduced the P321, also called the 3-in-1 Printer, a 24-pin printhead dot-matrix printer that retails for $699. The nickname 3-in-1 refers to the printer's combination of speed, letter quality type, and graphics. The quality of its output approaches that of laser printers which cost at least three times as much.

More Typeset Than Typewritten

Three different typefaces are built into the printer: Courier, Elite, and draft-quality. It can also hold two more typefaces in a cartridge and download one more from the computer, so its output is very flexible.

Draft mode is quite readable and extremely fast. The manufacturer claims 216 characters per second at 12 characters per inch and 180 cps at 10 cpi in this mode; letter quality runs at 72 cps. For listing computer programs or making quick dumps of a large amount of data, this printer performs extremely well. The Toshiba P321 can also produce proportionally spaced printing—allowing more space for wide letters like *w* than for narrow ones like *i*. When this feature is used, the printed output appears even and smooth, looking more like typeset-quality print than typewritten text.



This illustrates the graphics capabilities of the Toshiba P321.

```
:DRAW:CALL FRDM BASIC, 3 PARM
    S
;INCLUDED: # DF SHAPE,
;HPDS(PIXELS), VPOS(PIXELS)
;
DRAW LDA #2 ;ORG0
 STA NUMBYTES
;
 LDA #24 ;DRG1
 STA RDWCDUNT
;
 JSR ADDSHAPE ,WHICH SHAPE?
 BCS ERROR
 JSR GETHPDS ;WHERE?
 BCS ERROR
 JSR GETVPOS
 BCS ERROR
;
 LDA THISHAPE ;CDPY ADDR TO Z
    P
 STA PATTERN
 LDA THISHAPE+1
 STA PATTERN+1
;
DRAWLODP JSR CALC ;SCRN AODR
```

Draft mode on the P321.

The P321 can plot graphics with a resolution of 180 by 180 dots per inch. Although dots can be positioned with a horizontal resolution of 1/360 inch, two dots can't occupy adjacent positions. Unfortunately, it can't emulate Epson graphics. Epson was one of the first companies to offer an inexpensive printer that could produce graphics, and its graphics command set has since become an unofficial industry standard. While some newer software can produce graphics output for the P321, nearly every program that prints graphics can drive an Epson. If this feature had been included, Toshiba users would have enjoyed compatibility with a wider range of programs.

The Noise Factor

Laser printers are promoted as being quiet as a whisper. Naturally, the P321 isn't nearly that quiet. The noise level probably won't be offensive. But if you work in a quiet office, or if you compute at home and keep late hours, you might find the noise somewhat disturbing. It's not the loudest dot-matrix printer we've heard, but it might be loud enough to cause problems in some situations.

If you've previously been unsatisfied with near-letter-quality dot-matrix printers, the Toshiba P321 deserves consideration. The characters it produces look a little heavier than those made by a typewriter or a daisy wheel printer, but certainly better than the majority of dot-matrix printers we've seen. And when the Toshiba uses proportional spacing, its output looks better than what a typewriter could produce.

Toshiba P321 Printer
Toshiba America, Inc.
Information Systems Division
2441 Michelle Drive
Tustin, CA 92680
$699 (parallel only)
$749 (parallel and serial)
IBM Emulation Kit $49
Downloadable Type Font Kit $99

Murder On The Mississippi For Commodore And Apple

Kathy Yakal, Assistant Features Editor

Requirements: Commodore 64 or Apple II-series computer with at least 64K RAM. Joystick required. Disk only

Murder On The Mississippi, designed by Adam Bellin and published by Activision, is a rich, enjoyable adventure game. You're plunged into a convincing, complex world—a riverboat traveling down the Mississippi sometime in the 19th century. Though there is a lot to explore within that setting, it's not so huge and meandering that you get lost every time you make a move or have to keep retracing your steps. A cast of charming, eccentric characters makes you feel welcome in this imaginary world, and you *cannot* get killed five minutes into the game. In these and other ways, *Murder On The Mississippi* is free of the disagreeable aspects which reduce the fun of some other adventure games.

If you've ever played a poorly designed adventure game, the experience may have been frustrating enough to put you off the whole genre entirely. It seems that there are three areas in which many text-only or text-and-graphics adventures can miss the mark. First, some of them create a rather small world, or at least make it appear that way. As hard as you try, you can't get more than about ten minutes into the game without having to give up because you keep going around in circles. Second, some games have the nasty habit of allowing you to get into situations where you are easily killed, forcing you to start all over again. Finally, even if a game is playable, it may not have the feel of a real world. It's extremely difficult to create an environment and a set of characters with which you can easily and believably interact. And that is key to a good adventure game.

Trouble On The *Delta Princess*

On the other hand, a dedicated hardcore player of more traditional adventure games like Infocom's all-text *Zork* series may not find *Murder On The Mississippi* much of a challenge. Some people prefer to imagine what a game's world looks like, and aren't bothered by the hours it can take just to figure out how to move around and interact without getting killed. But for those who enjoy solving a murder mystery without bumping around in the dark, *Murder On The Mississippi* provides an entertaining, interactive environment in which to do just that.

As the player, you portray Sir Charles Foxworth, a famous British sleuth who is taking a three-day cruise down the Mississippi River on the *Delta Princess*. You are accompanied by your constant companion, Regis Phelps. While exploring the rooms on the ship, you come across a dead body and must enlist the help of passengers and crew members to find out who is the murderer. You have three days to solve the crime.

The game is entirely joystick-controlled; no keyboard commands are necessary. To move around the decks, to climb up and down stairs, and to enter rooms, you control the character by moving the joystick up, down, right, and left. It may take a few tries to maneuver your character into the exact spot that will make the door open, but it's not too tough.

The cabins themselves are not very big, so movement within them is rather restricted. If you're trying to get Sir Charles and Regis and a passenger to leave a room together, you sometimes get something of a Three Stooges effect—you keep bumping into each other as well as furniture and doors. But this tends to be amusing rather than irritating.

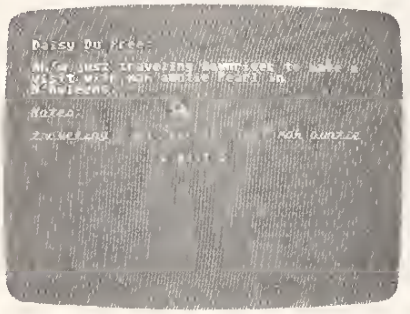
A Unique Interface

Adam Bellin has designed a unique user interface to allow interaction with the passengers. After you've entered a room, the character who resides there introduces himself or herself in response to your greeting. Pushing the joystick button will give you a menu: You can *Walk around, Inspect, Examine evidence, Talk to* (passenger's name), or return to the main menu. A small hand icon on the right side of the screen points to the selection highlighted, and pressing the joystick button activates that command.

If you choose to talk to the passenger, you're given another menu: *Tell me about, What do you know about this evidence?, Please follow me, Share notes with, Accuse,* or *Previous menu.* Information gathering is essential to solving the crime, so each passenger should be questioned, even if it leads nowhere. You can ask passengers to talk about themselves and about the victim.

After receiving information, Regis will ask if you'd like the notebook to take notes. If you think the information is important, you can choose to save certain key words from the passenger's speech. You're only allowed one line from each speech (generally 5–10 words), so choose carefully. Quite often, that's not enough, so you may want to take supplementary notes on paper That's a good idea in the beginning, anyway, as it will help you keep track of who's staying in which room.

As you select highlighted words to



be added to the notebook, an onscreen hand writes out the words in Sir Charles's handwriting. That's a nice touch, the kind of thing that surprises and delights a seasoned computer game player and makes computer games appealing to new users. *Murder On The Mississippi* contains many such thoughtful elements. Though the characters don't require a lot of depth in a game like this to make the game engaging, each is carefully drawn through the use of background, dialogue, and even accents. And Regis is an endearing fellow from the start—he's always following right on the heels of Foxworth, who appears to stand about two feet taller than his devoted sidekick.

Four Endings

In your early exploration of the ship, you'll discover that several of the rooms are locked. Finding out how to enter them merely takes some common sense, as does deciding what kind of evidence to pick up and keep for later examination. Getting to the point where you can actually start to draw some conclusions about the case will take some time and thought.

If you don't solve the mystery in one sitting (and you probably won't), you can save the game and later pick up where you left off. And there are four possible endings, so once you've solved the game, you can start over again and work your way through a new set of clues.

Murder On The Mississippi
Activision, Inc.
2350 Bayshore Frontage Road
Mountain View, CA 94043
$34.95 (Commodore)
$39.95 (Apple)

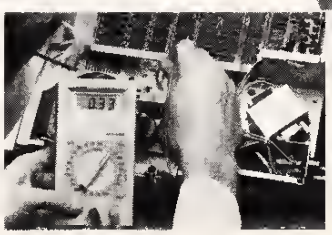
Only NRI teaches you to service and repair all computers as you build your own 16-bit IBM-compatible micro

As computers move into offices and homes by the millions, the demand for trained computer service technicians surges forward. The Department of Labor estimates that computer service jobs will actually *double* in the next ten years—a faster growth than any other occupation

Total System Training

As an NRI student, you'll get total hands-on training as you actually build your own Sanyo MBC-550 series computer from the keyboard up. Only a person who knows *all* the underlying fundamentals can cope with *all* the significant brands of computers. And as an NRI graduate, you'll possess the up-to-the-minute combination of theory and practical experience that will lead you to success on the job.

Your NRI course includes a Sanyo 16-bit microcomputer with 128K RAM, monitor, with double-density/double sided disk drive, and "Intelligent" Keyboard, The NRI Discovery Lab®, Teaching Circuit Design and Operations, a Digital Multimeter, Bundled Spread Sheet and Word Processing Software Worth over $1000 at Retail—and More



NRI is the only home study school that trains you as you assemble a top brand computer. You'll install and check keyboard power supply, disk drive and monitor, following step-by-step directions

You learn at your own convenience, in your own home, at your own comfortable pace. Without classroom pressures, without rigid night-school schedules, without wasted time. Your own personal NRI instructor and NRI's complete technical staff will answer your questions, give you guidance and special help whenever you may need it.

The Exciting Sanyo 16-bit IBM-compatible Computer—Yours To Keep

Critics hail the new Sanyo as the 'most intriguing' of all the IBM-PC compatible computers. It uses the same 8088 microprocessor as the IBM-PC and the MS/DOS operating system. So you'll be able to choose thousands of off-the-shelf software programs to run on your completed Sanyo.

As you build the Sanyo from the keyboard up, you'll perform demonstrations and experiments that will give you a total mastery of computer operations and servicing techniques. You'll do programming in BASIC language. You'll prepare interfaces for peripherals such as printers and joysticks. Using utility programs, you'll check out 8088 functioning. And the entire system, including all the bundled software and extensive data manuals, is yours to keep as part of your training.

100-Page Free Catalog Tells More

Send the coupon today for NRI's big 100-page color catalog, which gives you all the facts about NRI training in Microcomputers, Robotics, Data Communications, TV/Video/Audio Servicing, and other growing high-tech career fields. If the coupon is missing write to NRI at 3939 Wisconsin Ave., NW, Washington, DC 20016.

IBM is a Registered Trademark of International Business Machines Corporation

NRI SCHOOLS
McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue, Washington, DC 20016
We'll give you tomorrow.

For Career courses approved under GI bill, ☐ check for details

☑ **CHECK ONE FREE CATALOG ONLY**

- ☐ **Computer Electronics with Microcomputers**
- ☐ Data Communications
- ☐ Robotics & Industrial Controls
- ☐ Video Electronics Servicing
- ☐ Electronic Design Technology
- ☐ Digital Electronics
- ☐ Satellite Communications
- ☐ Communications Electronics
- ☐ Industrial Electronics
- ☐ Basic Electronics
- ☐ Telephone Servicing
- ☐ Small Engine Servicing
- ☐ Appliance Servicing
- ☐ Automotive Servicing
- ☐ Air Conditioning, Heating, Refrigeration, & Solar Technology
- ☐ Building Construction
- ☐ Locksmithing & Electronic Security

Name (Please Print) Age

Street

City/State/Zip Accredited by the National Home Study Council 198-086

Three Fantasy Games For Commodore And Apple

James V. Trunzo

Requirements: Commodore 64 or 128 (in 64 mode), or an Apple II-series computer with at least 64K RAM. Disk only.

The old axiom that good things come in threes certainly applies to a trio of new entertainment programs from Strategic Simulations, Inc. The wizards at SSI have conjured up three new fantasy titles that are sure to please all the would-be warriors who sit by their computers, anxious to explore another dungeon, slay another dragon, or banish another demon. And while on the surface it might appear to be unsound business strategy for a company to release three new monster and magic programs simultaneously, SSI succeeds because each game is unique in its approach and play. The three games, *Rings of Zilfin*, *Phantasie II*, and *Wizard's Crown*, will provide fantasy lovers with enough challenges to last the entire summer.

The first game, *Rings of Zilfin*, differs from other games of its kind by offering a nearly perfect hybrid of arcade action, role playing, and animation. The player controls a single character who has a variety of combat skills as well as latent magical ability. This ability must be developed during the course of the adventure in order to survive and complete your quest: You must reclaim the Rings of Zilfin and the fabulous Treasure of Fulgarsh.

Pay Attention To The Plants

The entire game is animated. Your keyboard-controlled character marches step by step across a huge mapped area. When he enters towns, dungeons, and so on, the program uses windowing to display the interiors and the options permitted. There's arcade-style combat as well.

But your character is not the typical warrior-hero. *Rings of Zilfin* requires a thorough understanding of strategy, economics, and diplomacy You need to pay special attention to mushrooms and plants because these items can offer important assistance. And, in addition to monsters, your persona will encounter elves, dwarfs, kings, beggars, witches, and wizards. Some are helpful, others deadly. Reading and rereading the well-written manual is a must; it contains necessary information as well as hidden clues.

This is a rich simulation. The realm of Batiniq contains three nations, 27 towns, two dungeons and more; there are over 100 inhabitants with whom to converse, and dozens of plants, magical pools, and monsters with which to contend—and all phases of the game are animated. The game has a flavor all its own. If you are a fantasy buff and you're looking for something a little different, *Rings of Zilfin* might be the game for you.

If you would enjoy something a little more traditional (and if you are one of the many who became addicted to the award-winning *Phantasie* game), you'll certainly want to get your hands on *Phantasie II*. The sequel does not require you to have played *Phantasie*, but if you have conquered the first *Phantasie* game, you can transfer your battle-trained characters to the new adventure.

Phantasie II has all the same features of its predecessor. Assembling a party of up to six characters, you must explore a vast wilderness, dungeons, Astral Planes and—new this time around—two levels of the Netherworld. Your group, made up of any mixture of fantasy types, must battle over 80 different monsters, gather treasure and magical artifacts, and improve its abilities as it attempts to defeat, once again, the arch-sorceror Nikademus.

The Hidden Undead

Phantasie II employs full screen graphics, animated combat, maze-like dungeons (which are mapped by the computer, incidentally), and a wide variety of terrains.

If you've played the original *Phantasie*, you can look forward to new features like molten lava, which is extremely dangerous; mist, which shrouds areas and hides such enemies as the undead and swarms of insects; and dark voids, which hold unknown horrors that must be faced by your group.

Also, a new wrinkle has been added to the combat phase of *Phantasie II*. Characters can now choose to toss rocks at enemies in any rank, with accuracy and damage determined by the appropriate skill level of the character.

A Most Unusual Game

The third game, *Wizard's Crown*, is the most difficult of the three and probably the most unusual fantasy game to hit the market in some time. Requiring 50 to 100 hours of playing time, *Wizard's Crown* comes very close in flavor to the actual *Dungeons and Dragons* role-playing game which started the fantasy craze. One reason for this is that each member of the party of adventurers can be controlled separately.

Also, the combat can be tactical in nature. Each character can select from 10 to 20 combat options, more than one in a given turn in most cases. For example, a warrior could improve his accuracy by aiming at an enemy prior to attacking. Characters can dodge and zigzag, attack defensively, stand on guard, load a bow or crossbow, move in any direction, or ready a new weapon—to name only a few of the options.

This control over individual movement allows the players to form a wide variety of defensive formations when in combat and also to take advantage of the battleground terrain. Because of the large number of combat variables that come into play—rear and flank attacks, for example—the combat is far closer to a typical war game than is usual in fantasy games.

Combat fought in the above manner can take anywhere from 10 to 20 minutes to complete, and all combat maneuvers are animated by highly detailed character icons. But if you're in a hurry, *Wizard's Crown* offers a quick combat option, too.

Especially Lifelike

Characters in *Wizard's Crown* have many more characteristics and skills than are usual in a game of this type. Combat awareness, ability to track, skill at adminstering first aid, knowing how to read ancient writings, and ability to use alchemy are some of the more esoteric ratings given characters in *Wizard's Crown*. These are in addition to the typical skills of a thief, wizard, or warrior. The various combination of skills add greatly to the personality and individuality of the characters, making them seem especially lifelike.

Your quest, to recover the coveted Wizard's Crown, takes your group of adventurers through streets, buildings, dense wilderness, and, of course, dungeons. During the course of your adventure, you will encounter dozens of monsters, find merchants with whom you can trade or sell your loot, bribe innkeepers for rumors and clues that will help you complete your quest, and acquire an almost limitless variety of magical items like lightning swords and rings of invisibility.

Wizard's Crown also includes five levels of difficulty, two kinds of combat, and works with one or two disk drives. Add this to all the other options, plus the excellent animation and graphics, and you have a game that will excite and challenge even the most seasoned

COMB

Authorized Liquidator

PUBLIC NOTICE

We are an authorized liquidator of consumer products that must be quickly sold at drastic price reductions for many reasons, such as Discontinued lines. Last year's models. Distributor's surplus. Cancelled orders. Bankruptcy. Factory overproducton. Out-of-season products. Tailends of inventories.

Whatever the reason for the liquidation you are assured of fine quality at money-saving low prices! Supplies are limited. Order now!

COMMODORE COMPATIBLE HARDWARE

SEE YOUR GAMES IN COLOR!

Save On A Factory New Discontinued Model

COMREX™ by EPSON®

13" COLOR MONITOR

- Liquidation of a discontinued model.
- Can be used with your VCR.
- Superior color picture quality.
- Manual controls for tint, color, contrast, horizontal, center and vertical hold.
- Power indicator light, video and sound input terminals.
- 270 line resolution.
- 40 character (8 x 8) x 24 line display.
- Size: 14" H x 13¾" W x 15½" D.
- Does not include connection cable. (Contact local stores.)

Compatible with Commodore 64K™ and Apple II® Series

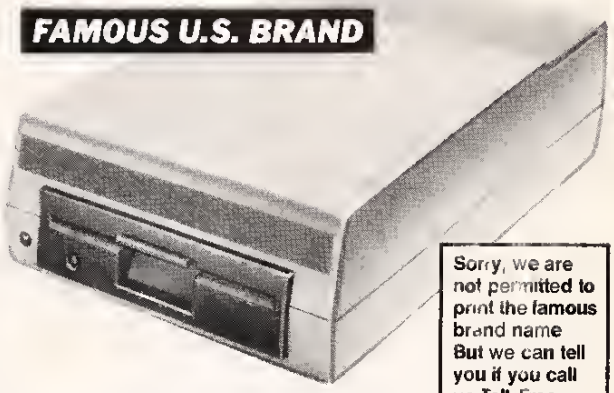
Limited Factory Warranty: 2 Years on Parts and Labor; 3 Years on Picture Tube

Mfr. List Price . $249.00

Liquidation Price **$149**

Item H-1382-7003-809
Shipping, handling: $12.00

FAMOUS U.S. BRAND



Sorry, we are not permitted to print the famous brand name. But we can tell you if you call us Toll-Free.

DISK DRIVE

Are you still wasting time using just ONE drive? Buy a SECOND drive to make quick back-up copies... and to use double-drive software with ease.

- Commodore compatible (C64, SX64).
- 2K RAM, 16K ROM. Maximum storage of 170K formatted data, 35 tracks.
- Uses 5¼" floppy diskettes.
- Serial interface.

These are factory reconditioned units, warranted by the U.S. manufacturer!

Original List Price . . $269.00

Liquidation Price . . . **$149**

Item H-1382-3553-013
Shipping, handling. $8.00

COMMODORE TRACTOR FEED

Liquidation is due to discontinued model. Now use continuous paper AND single sheets in your Commodore 803. Quickly snap in this tractor feed unit. When installed, you can still use your friction feed, too!

90-Day Limited Factory Warranty.

Mfr. List: $39.95 **$19**

Liquidation Price . .

Item H-1382-7004-286
Shipping, handling. $4.00

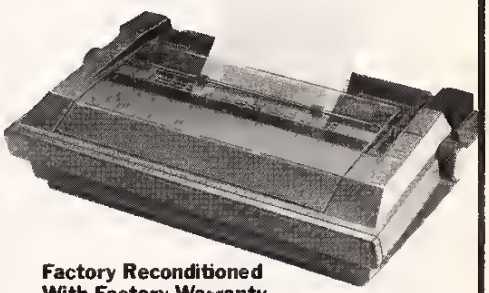
Famous Brand LETTER QUALITY PRINTER

- Commodore compatible (C64, SX64, Plus/4, C15, VIC20).
- Add to your present equipment for office quality appearance of letters, reports.
- Prints 18 cps. Bi-directional printing.
- Triumph-Adler compatible printwheel.
- Upper/lower case letters, numerals, symbols.
- Spacing: 10, 13, 15 cpi, or proportional selectable. 48 lines per inch maximum.
- 13" max. paper width (11" printable width).

Original List: $395.00 **$188**

Liquidation Price . . . . . .

Item H-1382-4184-016 Shipping, handling: $11.50



Factory Reconditioned With Factory Warranty.

C.O.M B. Direct Marketing Corp. Item H-1382
14605 28th Ave. N. / Minneapolis, MN 55441-3397

Send items shown at right. (Minnesota residents add 6% sales tax. Sorry, no C.O.D. orders.)

☐ My check or money order is enclosed. (No delays in processing orders paid by check, thanks to TeleCheck.)

Charge ☐ VISA® ☐ MasterCard® ☐ American Express®

Acct. No. _____ Exp. ___/___

PLEASE PRINT CLEARLY

Name _____

Address _____

City _____

State _____ ZIP _____

Phone (___) _____

Sign Here _____

SEND ME THE ITEMS I HAVE LISTED BELOW

Sales outside the 48 contiguous states are subject to special conditions. Please call or write to inquire.

Item No	#	Item	Price	S H

Credit card customers can order by phone, 24 hours a day, 7 days a week. VISA MasterCard AMERICAN EXPRESS

Toll-Free: 1-800-328-0609

TeleCheck Your check is welcome! No delays in orders paid by check.

Sales outside the 48 contiguous states are subject to special conditions. Please call or write to inquire.

COMB Direct Marketing Corp.
Authorized Liquidator
14605 28th Avenue North
Minneapolis, Minnesota 55441-3397

veteran of fantasy warfare.

SSI has created a triad of adventure games that offer something for everyone. Each program has its own special challenges and each requires a different strategy. One of them is sure to suit your taste; which one is up to you. You can't make a bad choice, though, because all three games are winners.

Rings of Zilfin
Phantasie II
Wizard's Crown
Strategic Simulations, Inc.
883 Stierlin Road
Mountain View, CA 94043
$39.95 each

Brattaccus

Charles Brannon, Program Editor

Requirements: Atari ST with color monitor, Commodore Amiga, or Apple Macintosh.

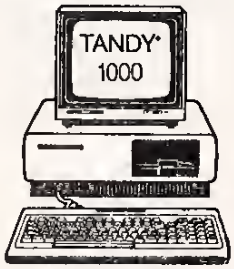
We've come a long way from the days of the original Adventure game. There are many variations in the genre of interactive fiction: text only, text and graphics, and graphics only. The text-only adventure games, best known by Infocom's *Zork* series and other interactive fiction such as *The Hitchhiker's Guide to the Galaxy*, depend on detailed prose and a sophisticated parser which decodes the typed commands you give to your invisible alter ego. To explore the adventure world, you type commands like GO WEST or TAKE ME TO YOUR LEADER. The game responds by changing the scene, giving you a new page of text to read, or responding with a message like CAN'T GO IN THAT DIRECTION, or CAN'T TAKE THE 'ME' The latter kind of message reveals the limitations of a command parser. The parser thinks you are trying to TAKE (pick up) the object ME.

This kind of adventure game can sometimes be frustrating, since only a limited number of actions make sense in any one scene. You are basically solving a series of linked or nested puzzles. For instance, you may start by trying to find a scroll that reveals the location of a magic key, which in turn opens the locked door that leads to the treasure you'll need to bribe a gatekeeper. In addition to a bribe, the gatekeeper may insist that you solve a knotty riddle before passing into the domain of a wizard who holds the ultimate object of your quest. Until you solve the gatekeeper's riddle, you can't enter that portion of the adventure world.

The text-only games make you feel

TANDY® COMPUTERS
SAVE 20—40%
Off List on All Tandy and RS Equipment

The IBM* PC compatible computer that's ahead of the crowd! Includes DeskMate® software for word processing, spreadsheet analysis, telecommunications and more, so you can use your computer right away. #25-1000
*IBM/TM International Business Machines Corp



Call For Latest Prices!

TAKE ADVANTAGE OF OUR MONTHLY SPECIALS ON COMPONENTS AND COMPLETE SYSTEMS

Call for monthly RS flyer specials.

Tandy Model 3000HD, List 3599. Our	2699.00	Model 6000 Multi user with XENIX,list 4499, our	CALL
Tandy Model 600, List 1599. Our	CALL	Tandy 1200HD, (XT compatible), List 1999. Our	CALL
Tandy Model 3000, List 2599. Our	CALL	Lotus 1-2-3. List 495. Our	CALL
Model 200 Portable. List 999. Our	CALL	Wordstar Professional. List 395, Our	CALL
Model 102 Portable. List 499, Our	CALL	DMP 130 Printer. List 350, Our	CALL
Non-RS Expansion Boards	CALL	EPSON Printers	CALL
Non-RS Hard Drives	CALL	All Tandy and RS monitors, peripherals	CALL

You will be pleased with our courteous, efficient service . . . and with the knowledge that we WILL NOT be undersold!

FOR COLORADO RESIDENTS AND INFORMATION CALL 303-249-9125

CALL TOLL FREE FOR ORDERS
1-800-44SHACK

AMERICAN EXPRESS · VISA · MasterCard

GREAT WESTERN ELECTRONICS
228 E. MAIN, MONTROSE, COLORADO 81401 HOURS Mon. - Fri. 9 a.m. - 5 p.m.

COMPUTE! Subscriber Services

Please help us serve you better. If you need to contact us for any of the reasons listed below, write to us at:

COMPUTE! Magazine
P.O. Box 10954
Des Moines, IA 50340

or call the Toll Free number listed below

Change Of Address. Please allow us 6–8 weeks to effect the change, send your current mailing label along with your new address

Renewal. Should you wish to renew your **COMPUTE!** subscription before we remind you to, send your current mailing label with payment or charge number or call the Toll Free number listed below

New Subscription. A one year (12 month) US subscription to **COMPUTE!** is $24.00 (2 years, $45.00; 3 years, $65.00. For subscription rates outside the US, see staff page) Send us your name and address or call the Toll Free number listed below

Delivery Problems. If you receive duplicate issues of **COMPUTE!**, If you experience late delivery or if you have problems with your subscription, please call the Toll Free number listed below.

COMPUTE!
1-800-247-5470
In IA 1-800-532-1272

LEARN PROGRAMMING

MASTER COMPUTERS IN YOUR OWN HOME
Now you can write programs and get a computer to do just what you want. Get the most out of any computer and avoid having to pay the high price of pre-packaged software

LEARN AT YOUR OWN PACE IN YOUR SPARE TIME
Our independent study program allows you to learn about computers, operations, applications and programming in your spare time at home. Our instructors provide you with one-on-one counseling

LEARN EVEN BEFORE YOU DECIDE ON A COMPUTER
Everything is explained in simple language. You will enjoy learning to use a computer—EVEN IF YOU DON'T OWN ONE. Learn to program on any personal computer; IBM, APPLE, COMMODORE, TRS, and more

BE YOUR OWN COMPUTER EXPERT
Programming is the best way to learn to use computers, and we can show you the best—and most economical—way to learn programming!
Send today for your free information package. No obligation. No salesman will call

halix CENTER FOR COMPUTER EDUCATION
INSTITUTE 1543 W. Olympic # 226 Los Angeles, CA 90015 3894

HALIX INSTITUTE CENTER FOR COMPUTER EDUCATION DEPT **61 8**
1543 W. OLYMPIC # 226 LOS ANGELES, CA 90015 3894

YES! Send me information on how I can learn about computers and programming at home!

Name _____ Age _____
Address _____
City _____ State/Zip _____

you're reading a complex novel in which you are the main character. You help "write" the story by making decisions at various branching points. However, there isn't enough room on the screen or in computer memory for both elaborate text and detailed color illustrations.

Adventure games that use both text and graphics rely on full-screen pictures to tell much of the story. While text-only games like *Zork* must describe a room, a graphics adventure like Sierra On-Line's *King's Quest* shows you the room itself, including whatever objects it contains. You still use keyboard commands to control the action, but the pictorial approach is one step closer to a real-life simulation.

Onscreen Alter Ego

Brattaccus is part of a new trend in adventure games in which you control a realistic image of a human or some other character. Instead of typing GO WEST, you move a mouse or a joystick, making your onscreen character walk around, open and close doors, pick up and put down objects, and even fight when necessary. In *Brattaccus*, the action takes place on a high-resolution stage of platforms, elevators, cantinas, police headquarters, and the criminal underworld. *Brattaccus* provides much of the interaction of text-only adventure games, but gives you direct, real-time control.

It takes some time to learn to control your character, a genetic engineer named Kyne. In addition to four basic directions, you can modify these four movements to get many more. In the Atari ST and Amiga versions, for example, you can push the mouse to the left, or push to the left with the right mouse button down, or even with both buttons pressed. Usually, your character behaves in a predictable fashion, but it can be frustrating to see him run and crash into a wall when you were merely trying to rotate to face a door.

In the game, Kyne has developed a new genetic technique for creating superhuman beings. The government, however, won't allow such a powerful, destabilizing technology to run rampant (at least, unless it controls the technology, with a race of supersoldiers foremost in mind). As Kyne, you have been falsely charged with selling your secrets to the underworld and are on the run, seeking out the seedy mining asteriod Brattaccus, where you believe you can find evidence to clear your name. The criminal underworld of Brattaccus is not unaware of the potential of your discoveries, so they too are hunting you. Fortunately, you are traveling under an alias, but there is a bartender who can blow your cover.

This would make for a great science-fiction film, and you become the star of the show. You walk Kyne's character around the maze of the asteroid's structure, wandering in and out of bars, floating up and down in elevators, moving from room to room, sometimes talking or fighting with other characters. Some characters let you know they are *going to the bar for a drink*, a cue for you to follow them for a private talk. These semi-autonomous characters roam throughout Brattaccus in rather aimless fashion. There are several classes of characters, from planetoid personnel and police to the henchmen of the criminal mastermind Kol Worpt.

Once in the bar, the characters ask if you'd like any information, usually in exchange for money or goods which are littered about the planetoid, ready to be plucked up by you or others. You respond to a character's prompt by choosing one of several responses that appear in a thought bubble above Kyne's head. Your choice affects the future of the game.

Realistic Swordplay

At times, you need to draw your sword to defend yourself against attackers. You can duck, parry, and lunge with your sword, but don't walk around with it drawn, since many characters take such behavior as a provocation. Characters whom you kill do not merely disappear, but instead lie on the ground for the rest of the game as a gruesome reminder. The game's graphics are realistic, and some players may object to this violent aspect.

Since many characters in *Brattaccus*—especially the police and henchmen—are excellent swordsmen, you'll find that games don't last long if you get carried away with swordplay. Swords, incidentally, are the only permissible weapons on Brattaccus, since other weapons could rupture the air bubble that keeps everyone alive on this desolate asteroid.

The world of *Brattaccus* is complex and difficult to map. In it you'll find security cameras that scan key corridors; you don't want to be caught fighting on camera. On/off switches let you control the operation of elevators, video screens, and more, but using them is a crime. Some rooms contain tannoys (loudspeakers) that periodically announce special police bulletins. Video screens display special news alerts. There are times where you'll be arrested and dragged off to jail, or captured by thugs and hauled away to an audience with the evil Kol Worpt. You must balance chit-chat, bribery, and measured doses of swordplay to keep things under control.

I don't know if *Brattaccus* is solvable. Although I've played it for weeks, it's still very hard to grasp all the elements needed to solve the puzzle and find the evidence. In this manner, *Brattaccus* is no different from other adventure games, which may take months to complete. For many people, this indicates good value, since the game still poses an interesting challenge even after considerable use.

Unrealized Potential

The only negative factors arise not from the game concept, but from its implementation. *Brattaccus* was first designed on the Atari 520ST, and the program's routines for moving the large objects representing characters can get bogged down when there are many characters on the screen at once. When *Brattaccus* was translated for the Amiga and Macintosh, apparently it was not rewritten to take advantage of these computers' features.

For instance, the Amiga's blitter chip, which could significantly speed up the animation, does not seem to be utilized to its potential. The game graphics are absolutely identical on both machines. And curiously, though the Amiga works with the same type of joystick as the ST, joystick control is absent from the Amiga version. Also, the Amiga version makes no use of the Amiga's integral speech synthesis.

The Macintosh version's graphics are somewhat disappointing, too. The designers converted the ST color graphics without taking advantage of the greater vertical resolution on the Mac. As a result, the Macintosh version has only 200 lines of vertical resolution and looks squashed compared to the original.

Nevertheless, *Brattaccus* shows the possibilities for gameplay on powerful 68000-based computers such as the ST, Amiga, and Macintosh. As designers continue to learn more about these machines, we can expect new waves of entertainment software which take advantage of the powerful CPU, large-capacity disks, digital sound, and elaborate screen graphics that make these computers so attractive.

Brattaccus
Mindscape, Inc.
3444 Dundee Road
Northbrook, IL 60062
$49.95

©

SPRITE 32
For Commodore 64

Jeremy Zullo

This sophisticated utility allows the Commodore 64 to display as many as 32 sprites on the screen at the same time. (It also works on the Commodore 128 in 64 mode.) The "Sprite BASIC" enhancement program adds several new sprite commands to BASIC 2.0. For machine language programmers, the "Sprite Kernal" utility offers the same capabilities for ML programming. Demonstration programs show how to use the technique in both BASIC and machine language. A disk drive is required.

You probably know that the Commodore 64 is designed to display a maximum of eight sprites on the screen at one time. That's enough for most purposes, but there are many situations, particularly in game programming, where extra sprites would be useful. The programs accompanying this article let you display as many as 32 sprites on the screen at once. Though the programs are written in machine language, you can use them without being a machine language expert.

"Sprite 32" is the first program you'll need; it handles the mechanics of displaying the extra sprites. The second utility, a BASIC enhancement called "Sprite BASIC," adds nine new sprite commands to the 64's BASIC 2.0. The third ML program, called "Sprite Kernal," offers a convenient way for machine language programmers to access all of the Sprite 32 functions.

Getting Started

Begin by entering Programs 1, 2, and 3. Because these programs are written in machine language, you must enter them with the "MLX" machine language entry program listed elsewhere in this issue. Before you type in the programs, read the information below about which filenames to use when saving them. If you don't intend to program in machine language, you need not type in Program 3; however, you may want to enter it anyway to view the machine language demonstration (see below). Here are the addresses you need to enter each program with MLX:

Program 1

Starting address: C000
Ending address: C4C7

Program 2

Starting address: C600
Ending address: C997

Program 3

Starting address: C600
Ending address: C80F

If you wish to use the demo programs included with this article, you must save Programs 1, 2, and 3 with the exact filenames listed here:

Program 1: **SPRITE 32**
Program 2: **SPRITE BASIC**
Program 3: **SPRITE KERNAL**

After you've saved Programs 1–3, you may want to enter Program 4, the BASIC demonstration. *Before* entering this program, however, you must activate Sprite BASIC. Load the program with the command LOAD"SPRITE BASIC",8,1. When the load is finished, enter NEW to reset the computer's memory pointers, then type SYS 50688 and press RETURN to install Sprite BASIC. *It is very important that you install Sprite BASIC before typing in Program 4.* If you omit this step, the program will not work correctly.

After Sprite BASIC is installed, enter Program 4. Don't worry about the unfamiliar commands; they'll be explained in the next section. Save a copy of the program, then run it. After loading Sprite 32 and Sprite BASIC, the program displays 32 sprites on the screen, LISTs itself, and returns to ready mode.

Note that Sprite 32 works completely in the background: The sprites remain stable even after the READY prompt and blinking cursor reappear. You can LIST the program, edit it, and with one exception (see below) use BASIC in the normal way.

BASIC Demo

Let's examine some Sprite BASIC commands. With 32 sprites still on the screen, type this statement and press RETURN:

```
SPRITE 0
```

All of the sprites disappear. Now enter the command SPRITE 1: All of the sprites instantly reappear.

The SPRITE command turns the Sprite 32 utility on and off. This command is important because *you must always disable Sprite 32 before using the disk or tape drive.* If you try to save or load a program while Sprite 32 is still active, you may crash the system (no harm is done to the computer, but you might lose whatever program is in memory).

Here are some additional commands to try. Type in each of the lines listed here, pressing RETURN at the end of each line:

```
FOR J=0 TO 7:DISABLE 3,J:NEXT
ENABLE 3,0
FOR J=1 TO 7:ENABLE 3,J:NEXT
```

The ENABLE and DISABLE commands let you turn individual sprites on and off. The first number after the command indicates the sprite's group number. There are four sprite groups, numbered 0–3. Each group contains eight sprites, and group 0 is always located at the top of the screen. Within each group, sprites are numbered from 0–7; in this demo, sprite 0 is at the leftmost screen position.

The second parameter in the ENABLE and DISABLE commands identifies which sprite within the group you wish to affect. Thus, DISABLE 3,0 turns off sprite 0 in

group 3 (the bottom group). ENABLE 2,7 turns on the rightmost sprite in group 2, and so on.

Horizontal Zones

Sprite 32 divides the screen horizontally into four separate zones, one for each group of eight sprites. When all 32 sprites are on the screen, each group is confined to its own horizontal zone. For example, you cannot move a group 3 sprite into the zone for group 2. However, by sacrificing sprites from other zones, you can allow a sprite to move freely through two or more zones. The basic method is to DISABLE the corresponding sprite in the next higher-numbered zone.

For instance, if you disable sprite 0 in group 3, then sprite 0 in group 2 can move anywhere within zones 2 and 3. By sacrificing three corresponding sprites, you can allow a sprite from group 0 to go anywhere on the screen. To illustrate, enter these lines, pressing RETURN at the end of each line:

```
FOR J=1 TO 3:DISABLE J,0:NEXT
FOR J=60 TO 250:PLACE 0,0,30,J:NEXT
FOR J=250 TO 60 STEP -1:PLACE
  0,0,30,J:NEXT
```

Sprite 0 from group 0 moves all the way down through zones 1, 2, and 3, then returns to its original position. While this method reduces the total number of sprites you can display, it does permit you to have some sprites that aren't confined to particular screen areas.

One word of warning: Do not disable any of the sprites in group 0, or you may get unpredictable results.

Sprite BASIC Commands

Here is a list of all the Sprite BASIC commands:

DISABLE ***sprite group,sprite number*** Turn off a sprite. The *sprite group* parameter can range from 0–3 and identifies which of four groups the sprite belongs to. The *sprite number* can range from 0–7 and identifies an individual sprite within the group (see above).

ENABLE ***sprite group,sprite number*** Turn on the sprite specified by *sprite group* and *sprite number* (see above).

KILL Deactivate Sprite BASIC. After you perform KILL, Sprite BASIC is disabled and the 64's BASIC works exactly as usual. This is not the same as a SPRITE 0 statement (see below), which disables the Sprite 32 utility but does not affect Sprite BASIC.

OFF ***sprite group,sprite number*** Make the designated sprite invisible. Use the PUTS command (see below) to make a sprite visible again. Note the difference between OFF and DISABLE: An OFF statement makes the sprite disappear from the screen but has no effect on the ability of other sprites to venture into that sprite's zone. A DISABLE statement allows another sprite to move through the disabled sprite's territory and also makes the sprite disappear.

PLACE ***sprite group,sprite number,X coord,Y coord*** Place the designated sprite at the screen coordinates indicated by *X coord* and *Y coord*. The horizontal coordinate *X coord* can be any value from 0–512, but only coordinates from 24–343 are visible on the screen. The vertical coordinate *Y coord* can be any value from 0–255, but only coordinates from 50–249 are visible on the screen. (No special tricks are required to move sprites past the "seam" into horizontal positions greater than 255; Sprite 32 automatically handles the most significant bit for horizontal positioning.)

PUTS ***sprite group,sprite number*** The opposite of OFF, this statement makes a sprite visible.

RASTL ***boundary number,new raster*** The RASTL (RASTer Line) statement lets you change the boundary between two sprite zones; since the zones are contiguous, this also changes the size of those zones. The first parameter, *boundary number*, identifies which zone boundary you wish to change. There are three boundaries, numbered 0–2, which separate the four sprite zones. Boundary 0 separates zones 0 and 1; boundary 1 separates zones 1 and 2; and boundary 2 separates zones 2 and 3.

The second parameter, *new raster*, specifies the raster line where the specified boundary should be located. The visible screen contains 200 raster lines, numbered 50–249, with line 50 at the very top of the screen. The default position for boundary 0 is raster line 99. To move this boundary 20 lines higher on the screen (to line 79), use the statement RASTL ,0,79. Now the lower portion of zone 0 ends at screen line 79 and the upper portion of zone 1 begins at line 80.

SET ***sprite group,sprite number, shape,color*** SET defines the shape and color of the individual sprite specified by *sprite group* and *sprite number*. The *shape* parameter tells the 64 where to find the shape data for the sprite. This is the same value you would POKE into one of the shape pointer locations from 2040– 2047 under normal circumstances. The *color* value can range from 0–15 and corresponds to the usual 64 color numbers (color 0 is black, and so forth). Your user's manual contains more information about colors and sprite shape pointers.

SPRITE ***toggle*** Turn Sprite 32 on or off. Because Sprite 32 interferes with disk and tape operations (including saving and loading programs), you must *always* turn it off before using disk or tape. Use SPRITE 0 to deactivate Sprite 32, and SPRITE 1 to activate it. This statement does not affect Sprite BASIC, which must always be active in order to use a program that contains Sprite BASIC commands. For instance, after loading Sprite BASIC into memory, Program 4 activates it with SYS 50688 before performing any Sprite BASIC commands.

Programming Tips

When placing sprites on the screen, keep in mind that no part of the sprite can cross the boundaries of its zone unless you have DISABLEd other sprites to permit multizone movement.

For example, the default location for zone 0 is from raster lines 0–99. Since a sprite can be as many as 21 lines high, you should not attempt to PLACE a group 0 sprite using a vertical coordinate greater than 78 (99−21=78). Similarly, zone 1 stretches from lines 100–149, so a zone 1 sprite can move between lines 100–128 (149−21=128). If you try to position a sprite outside its permitted zone, it may flicker or disappear completely. Within its horizontal

zone, a sprite can have any horizontal location.

There are certain aspects of sprite behavior which Sprite 32 doesn't affect at all. For instance, sprite-to-sprite display priorities are exactly the same as usual: When two or more sprites overlap, lower-numbered sprites always appear in front of higher-numbered ones.

You may change the sprite-to-background priority of a sprite in the usual way, but the change affects every sprite of the same number. That is, if you change the sprite/background priority for sprite 0, it is changed for sprite 0 in every sprite group.

The same is true of horizontal or vertical expansion. Expansion affects every like-numbered sprite on the screen.

Machine Language Demo

For machine language programming, BASIC commands are not particularly convenient. Program 3, the Sprite Kernal, provides all the features of Sprite 32 to machine language programmers. Even if you don't understand machine language, you may want to enter and run the remaining programs to see an impressive demonstration. Program 5 illustrates the power of machine language by moving 17 sprites on the screen simultaneously. This program must be entered with MLX, using these addresses:

Starting address: 6000
Ending address: 62B7

If you have been using Sprite 32 or Sprite BASIC, turn the computer off and on before you load and run MLX. Be sure to save Program 5 with the filename ML DEMO.

Next, type in and save Program 6 (you do not have to install Sprite BASIC before typing this program). This is a short BASIC loader that installs the necessary ML programs in memory, then starts ML DEMO with the statement SYS 24576.

When you run Program 6, the screen fills immediately with 17 bouncing sprites. Note that several of the sprites move through more than one sprite zone; one of them, the light blue sprite, is able to move anywhere on the screen. As explained earlier, it is necessary to sacrifice a certain number of sprites to achieve this effect.

"Sprite 32" allows the Commodore 64 to display as many as 32 sprites on the screen simultaneously.

Press RUN/STOP–RESTORE to stop the program. To restart it, enter SYS 24576.

The Sprite Kernal

Like Sprite BASIC, the Sprite Kernal also requires that Sprite 32 be in memory. Here are the starting addresses for each Sprite Kernal routine:

Routine	JSR address
SPRITE	$C612/50706
PLACE	$C615/50709
SET	$C618/50712
OFF	$C61B/50715
PUTS	$C61E/50718
DISABLE	$C621/50721
ENABLE	$C624/50724
RASTL	$C627/50727

The Sprite Kernal routines perform the same functions as their Sprite BASIC equivalents. However, a different procedure is used to pass each routine the information it needs. The basic method is to store the parameters in memory locations beginning at 50688 ($C600), then call the Sprite Kernal routine with JSR. For an explanation of the parameters required by each routine, see "Sprite BASIC Commands" above.

Since the SPRITE routine takes only one parameter (1 or 0), you need to supply only one value before calling it. For example, to perform the equivalent of the Sprite BASIC statement SPRITE 1, you would execute LDA #1:STA $C600: JSR $C612. To do the equivalent of SPRITE 0, use LDA #0:STA $C600: JSR $C612. All of the remaining Sprite Kernal routines require two or more parameters. Here is an outline of how to call them:

PLACE ($C615/50709) Store the *sprite group* value in $C600/50688 and the *sprite number* value in $C601/50689. Locations $C602–$C603/50690–50691 hold the low byte and high byte of the sprite's horizontal (X) position. Store the sprite's vertical (Y) position in location $C604/50692.

SET ($C617/50712) Store the *sprite group* value in $C600/50688 and the *sprite number* value in $C601/50689. Store the *shape* pointer value in $C602/50690 and the *color* value in $C603/50691.

OFF ($C61B/50715) Only two values are required. Store the *sprite group* value in $C600/50688 and the *sprite number* value in $C601/50689.

PUTS ($C61E/50718) The converse of OFF. Store the *sprite group* value in $C600/50688 and the *sprite number* value in $C601/50689.

DISABLE ($C621/50721) Only two values are required. Store the *sprite group* value in $C600/50688 and the *sprite number* value in $C601/50689.

ENABLE ($C624/50724) The converse of DISABLE. Store the *sprite group* value in $C600/50688 and the *sprite number* value in $C601/50689.

RASTL ($C627/50727) Store the *boundary number* value in $C600/50688 and the *new raster* value in $C601/50689.

Here is a short example of how to use Sprite Kernal routines. This program displays sprite 4 in group 2. You will need a machine language assembler to create the object code for this routine. The comments following the semicolons are optional and need not be included.

```
LDA   #$01 ;turn on
STA   $C600 ;Sprite 32
JSR   $C612
LDA   #$04 ;sprite number
STA   $C600
LDA   #$02 ;sprite group
STA   $C601
LDA   #$A0 ;low and high
STA   $C602 ;bytes of the
LDA   #$00 ;sprite's
STA   $C603;X coordinate
LDA   #60
STA   $C604 ;Y coordinate
JSR   $C615 ;PLACE
RTS
```

When Sprite 32 is active, the 64's IRQ vector is diverted from its normal address to the custom routines used to display extra sprites. If

you activate another interrupt-driven routine at the same time, the conflict may produce unexpected results.

Program 1: Sprite 32

Please refer to the "MLX" article in this issue before entering the following listing.

```
C000:4C A7 C0 00 00 00 00 00 A9
C008:00 00 00 00 45 45 45 45 98
C010:45 45 45 45 00 01 02 03 8D
C018:04 05 06 07 00 01 02 03 19
C020:04 05 06 07 00 00 00 00 16
C028:00 00 00 00 00 77 77 77 ED
C030:77 77 77 77 77 00 01 02 71
C038:03 04 05 06 07 00 01 02 79
C040:03 04 05 06 07 00 00 00 7D
C048:00 00 00 00 00 00 A9 A9 C6
C050:A9 A9 A9 A9 A9 A9 00 01 D5
C058:02 03 04 05 06 07 00 01 B9
C060:02 03 04 05 06 07 00 00 C0
C068:00 00 00 00 00 00 00 DB C5
C070:DB DB DB DB DB DB D8 00 16
C078:01 02 03 04 05 06 07 00 E9
C080:01 02 03 04 05 06 07 01 F2
C088:01 01 01 01 01 01 01 01 0A
C090:01 01 01 01 01 01 01 01 12
C098:01 01 01 01 01 01 01 FF 19
C0A0:FF FF FF 64 96 C8 FA 78 AE
C0A8:A9 1B 8D 11 D0 A9 81 8D 46
C0B0:1A D0 A9 C3 8D 14 03 A9 51
C0B8:C0 8D 15 03 A9 7F 8D 0D 44
C0C0:DC 58 60 A9 01 8D 19 D0 AE
C0C8:AD 12 D0 CD A3 C0 90 03 E0
C0D0:4C A8 C1 AD 0B C0 8D 10 3C
C0D8:D0 AD 9F C0 8D 15 D0 AD 3E
C0E0:03 C0 8D 00 D0 AD 0C C0 DB
C0E8:8D 01 D0 AD 14 C0 8D 27 4C
C0F0:D0 AD 1C C0 8D F8 07 AD E1
C0F8:04 C0 8D 02 D0 AD 0D C0 96
C100:8D 03 D0 AD 15 C0 8D 28 EE
C108:D0 AD 1D C0 8D F9 07 AD 1F
C110:05 C0 8D 04 D0 AD 0E C0 52
C118:8D 05 D0 AD 16 C0 8D 29 90
C120:D0 AD 1E C0 8D FA 07 AD 58
C128:06 C0 8D 06 D0 AD 0F C0 0D
C130:8D 07 D0 AD 17 C0 8D 2A 32
C138:D0 AD 1F C0 8D FB 07 AD 97
C140:07 C0 8D 08 D0 AD 10 C0 C7
C148:8D 09 D0 AD 18 C0 8D 28 D3
C150:D0 AD 20 C0 8D FC 07 AD D3
C158:08 C0 8D 0A D0 AD 11 C0 82
C160:8D 0B D0 AD 19 C0 8D 2C 75
C168:D0 AD 21 C0 8D FD 07 AD 10
C170:09 C0 8D 0C D0 AD 12 C0 3D
C178:8D 0D D0 AD 1A C0 8D 2D 17
C180:D0 AD 22 C0 8D FE 07 AD 4C
C188:0A C0 8D 0E D0 AD 13 C0 F7
C190:8D 0F D0 AD 1B C0 8D 2E B8
C198:D0 AD 23 C0 8D FF 07 AD 88
C1A0:A3 C0 8D 12 D0 4C B6 C4 E2
C1A8:CD A4 C0 90 03 4C AD C2 C4
C1B0:AD 2C C0 8D 10 D0 AD A0 C6
C1B8:C0 8D 15 D0 AD 87 C0 F0 AD
C1C0:18 AD 24 C0 8D 00 D0 AD 08
C1C8:2D C0 8D 01 D0 AD 35 C0 3D
C1D0:8D 27 D0 AD 3D C0 8D F8 DA
C1D8:07 AD 88 C0 F0 18 AD 25 D0
C1E0:C0 8D 02 D0 AD 2E C0 8D AA
C1E8:03 D0 AD 36 C0 8D 28 D0 98
C1F0:AD 3E C0 8D F9 07 AD 89 9C
C1F8:C0 F0 18 AD 26 C0 8D 04 4A
C200:D0 AD 2F C0 8D 05 D0 AD 1B
C208:37 C0 8D 29 D0 AD 3F C0 1A
C210:8D FA 07 AD 8A C0 F0 18 28
C218:AD 27 C0 8D 06 D0 AD 30 2E
C220:C0 8D 07 D0 AD 38 C0 8D B4
C228:2A D0 AD 40 C0 8D FB 07 EB
C230:AD 8B C0 F0 18 AD 28 C0 1F
C238:8D 08 D0 AD 31 C0 8D 09 2C
C240:D0 AD 39 C0 8D 2B D0 AD 35
C248:41 C0 8D FC 07 AD 8C C0 E8
C250:F0 18 AD 29 C0 8D 0A D0 BD
C258:AD 32 C0 8D 0B D0 AD 3A 63
C260:C0 8D 2C D0 AD 42 C0 8D C1
C268:FD 07 AD 8D C0 F0 18 AD E4
C270:2A C0 8D 0C D0 AD 33 C0 12
C278:8D 0D D0 AD 3B C0 8D 2D 22
C280:D0 AD 43 C0 8D FE 07 AD 72
C288:8E C0 F0 18 AD 28 C0 8D 4E
C290:0E D0 AD 34 C0 8D 0F D0 75
C298:AD 3C C0 8D 2E D0 AD 44 49
C2A0:C0 8D FF 07 AD A4 C0 8D 69
C2A8:12 D0 4C B6 C4 CD A5 C0 C9
C280:90 03 4C B2 C3 AD 4D C0 24
C2B8:8D 10 D0 AD A1 C0 8D 15 3E
C2C0:D0 AD 8F C0 F0 18 AD 45 A0
C2C8:C0 8D 00 D0 AD 4E C0 8D D4
C2D0:01 D0 AD 56 C0 8D 27 D0 81
C2D8:AD 5E C0 8D F8 07 AD 90 8D
C2E0:C0 F0 18 AD 46 C0 8D 02 33
C2E8:D0 AD 4F C0 8D 03 D0 AD FF
C2F0:57 C0 8D 28 D0 AD 5F C0 43
C2F8:8D F9 07 AD 91 C0 F0 18 09
C300:AD 47 C0 8D 04 D0 AD 50 30
C308:C0 8D 05 D0 AD 58 C0 8D DE
C310:29 D0 AD 60 C0 8D FA 07 55
C318:AD 92 C0 F0 18 AD 48 C0 0B
C320:8D 06 D0 AD 51 C0 8D 07 94
C328:D0 AD 59 C0 8D 2A D0 AD 1F
C330:61 C0 8D FB 07 AD 93 C0 E0
C338:F0 18 AD 49 C0 8D 08 D0 A5
C340:AD 52 C0 8D 09 D0 AD 5A 65
C348:C0 8D 2B D0 AD 62 C0 8D 0C
C350:FC 07 AD 94 C0 F0 18 AD BE
C358:4A C0 8D 0A D0 AD 53 C0 2C
C360:8D 0B D0 AD 5B C0 8D 2C 88
C368:D0 AD 63 C0 8D FD 07 AD 5C
C370:95 C0 F0 18 AD 48 C0 8D 3C
C378:0C D0 AD 54 C0 8D 0D D0 5C
C380:AD 5C C0 8D 2D D0 AD 64 53
C388:C0 8D FE 07 AD 96 C0 F0 5E
C390:18 AD 4C C0 8D 0E D0 AD 19
C398:55 C0 8D 0F D0 AD 5D C0 56
C3A0:8D 2E D0 AD 65 C0 8D FF B8
C3A8:07 AD A5 C0 8D 12 D0 4C 82
C3B0:B6 C4 CD A6 C0 90 03 4C 83
C3B8:B1 C4 AD 6E C0 8D 10 D0 14
C3C0:AD A2 C0 8D 15 D0 AD 97 97
C3C8:C0 F0 18 AD 66 C0 8D 00 1C
C3D0:D0 AD 6F C0 8D 01 D0 AD E5
C3D8:77 C0 8D 27 D0 AD 7F C0 6D
C3E0:8D F8 07 AD 98 C0 F0 18 EA
C3E8:AD 67 C0 8D 02 D0 AD 70 31
C3F0:C0 8D 03 D0 AD 78 C0 8D 08
C3F8:28 D0 AD 80 C0 8D F9 07 BD
C400:AD 99 C0 F0 18 AD 68 C0 F6
C408:8D 04 D0 AD 71 C0 8D 05 FC
C410:D0 AD 79 C0 8D 29 D0 AD 09
C418:81 C0 8D FA 07 AD 9A C0 D8
C420:F0 18 AD 69 C0 8D 06 D0 8D
C428:AD 72 C0 8D 07 D0 AD 7A 67
C430:C0 8D 2A D0 AD 82 C0 8D 56
C438:FB 07 AD 9B C0 F0 18 AD 98
C440:6A C0 8D 08 D0 AD 73 C0 46
C448:8D 09 D0 AD 78 C0 8D 2B F4
C450:D0 AD 83 C0 8D FC 07 AD 46
C458:9C C0 F0 18 AD 6B C0 8D 2A
C460:0A D0 AD 74 C0 8D 0B D0 43
C468:AD 7C C0 8D 2C D0 AD 84 5D
C470:C0 8D FD 07 AD 9D C0 F0 44
C478:18 AD 6C C0 8D 0C D0 AD FE
C480:75 C0 8D 0D D0 AD 7D C0 70
C488:8D 2D D0 AD 85 C0 8D FE 62
C490:07 AD 9E C0 F0 18 AD 6D 99
C498:C0 8D 0E D0 AD 76 C0 8D 0B
C4A0:0F D0 AD 7E C0 8D 2E D0 EC
C4A8:AD 86 C0 8D FF 07 4C B1 02
C4B0:C4 A9 00 8D 12 D0 AD 0D 1C
C488:DC 29 01 F0 03 4C 31 EA C0
C4C0:4C BC FE 00 00 00 00 00 7F
```

Program 2: Sprite BASIC

Please refer to the "MLX" article in this issue before entering the following listing.

```
C600:A2 07 BD 04 03 9D A7 02 78
C608:BD 12 C6 9D 04 03 CA 10 7D
C610:F1 60 5F C6 E3 C6 18 C7 39
C618:3A C7 4B 49 4C CC 53 45 34
C620:D4 4F 46 46 D3 50 55 54 F7
C628:D3 44 49 53 41 42 4C C5 80
C630:45 4E 41 42 4C C5 50 4C A6
C638:41 43 C5 52 41 53 54 CC E1
C640:53 50 52 49 54 C5 4C 4F 0C
C648:4F CB 00 8C C7 83 C8 BD D4
C650:C8 E7 C8 0E C9 32 C9 A7 88
C658:C7 56 C9 6C C9 6C C7 20 0F
C660:7C A5 A2 00 A0 04 84 0F 17
C668:BD 00 02 85 08 C9 22 F0 0A
C670:4F 24 0F 70 26 C9 41 90 03
C678:22 C9 5B B0 1E 84 71 A0 86
C680:4C 84 0B A0 FF 86 7A CA 9A
C688:C8 E8 BD 00 02 38 F9 1A 6B
C690:C6 F0 F5 C9 80 D0 30 05 C5
C698:0B A4 71 E8 C8 99 FB 01 37
C6A0:89 FB 01 F0 36 38 E9 3A D9
C6A8:F0 04 C9 49 D0 02 85 0F 26
C680:38 E9 55 D0 B3 85 08 BD 0E
C688:00 02 F0 DF C5 08 F0 DB EE
C6C0:C8 99 F8 01 E8 D0 F0 A6 BB
C6C8:7A E6 0B C8 89 19 C6 10 0B
C6D0:FA B9 1A C6 D0 B4 BD 00 CE
C6D8:02 10 BE 99 FD 01 A9 FF 24
C6E0:85 7A 60 10 2A C9 FF F0 46
C6E8:26 24 0F 30 22 C9 CC 90 D9
C6F0:24 38 E9 CB AA 84 49 A0 33
C6F8:FF CA F0 08 C8 B9 1A C6 FF
C700:10 FA 30 F5 C8 B9 1A C6 E3
C708:30 08 20 47 AB D0 F5 4C 03
C710:F3 A6 4C EF A6 4C 1A A7 0E
C718:20 73 00 C9 CC 90 15 20 24
C720:25 C7 4C AE A7 E9 CC 0A 31
C728:A8 89 4C C6 48 89 48 C6 F6
C730:48 4C 73 00 20 79 00 4C 98
C738:E7 A7 A9 00 85 0D 20 73 EE
C740:00 C9 FF F0 21 C9 D5 90 BD
C748:1D 38 E9 D5 0A 48 20 73 34
C750:00 20 F1 AE 68 A8 89 5D C7
C758:C6 85 55 AD 5E C6 85 56 A1
C760:20 54 00 4C 8D AD 20 79 B6
C768:00 4C 8D AE A5 15 48 A5 5F
C770:14 48 20 F7 B7 A0 00 B1 91
C778:14 85 63 C8 B1 14 85 62 87
C780:68 85 14 68 85 15 A2 90 05
C788:38 20 49 BC 60 A2 07 BD 8A
C790:A7 02 9D 04 03 CA 10 F7 C3
C798:60 00 00 00 00 00 00 00 58
C7A0:01 02 04 08 10 20 40 80 34
C7A8:20 8A AD 20 F7 B7 A5 14 A0
C7B0:8D 9A C7 20 F1 B7 8E 99 CD
C7B8:C7 20 FD AE 20 EB B7 A5 A4
C7C0:14 8D 9B C7 A5 15 8D 9C E6
C7C8:C7 8E 9D C7 AD 9A C7 18 8F
C7D0:0A 0A 0A 0A 6D 9A C7 CC
C7D8:6D 99 C7 AA 8E 9E C7 AD 55
C7E0:9B C7 9D 03 C0 AD 9E C7 D5
C7E8:38 ED 99 C7 AA AC 99 C7 C2
C7F0:AD 9C C7 F0 41 B9 A0 C7 80
C7F8:8D 9F C7 BD 0B C0 19 A0 3A
C800:C7 9D 0B C0 8A 18 69 21 F2
C808:AA AD 9A C7 18 0A 0A 0A 31
C810:6D 99 C7 A8 E0 84 B0 1B D7
C818:B9 87 C0 D0 16 BD 0B C0 0C
C820:0D 9F C7 9D 0B C0 8A 18 7B
C828:69 21 AA 98 18 69 08 A8 84
C830:4C 14 C8 4C 7A C8 AC 99 B4
C838:C7 A9 FF 38 F9 A0 C7 8D 0B
C840:9F C7 BD 0B C0 2D 9F C7 BD
C848:9D 0B C0 8A 18 69 21 AA 7F
C850:AD 9A C7 18 0A 0A 0A 6D D3
C858:99 C7 A8 E0 84 B0 1B B9 A2
C860:87 C0 D0 16 BD 0B C0 2D 2A
C868:9F C7 9D 0B C0 8A 18 69 E9
```

```
C870:21 AA 98 18 69 08 A8 4C DA
C878:5B C8 AE 9E C7 AD 9D C7 A1
C880:9D 0C C0 60 20 8A AD 20 A8
C888:F7 B7 A5 14 8D 9A C7 20 80
C890:F1 B7 8E 99 C7 20 F1 B7 CE
C898:8E 9B C7 20 F1 B7 8E 9C 78
C8A0:C7 AD 9A C7 18 0A 0A 0A 58
C8A8:0A 0A 6D 9A C7 6D 99 C7 08
C8B0:AA AD 98 C7 9D 1C C0 AD 7F
C8B8:9C C7 9D 14 C0 60 20 8A D1
C8C0:AD 20 F7 B7 A5 14 8D 9A DE
C8C8:C7 20 F1 B7 8E 99 C7 AC 17
C8D0:99 C7 A9 FF 38 F9 A0 C7 09
C8D8:8D 9B C7 AC 9A C7 89 9F E2
C8E0:C0 2D 9B C7 99 9F C0 60 3B
C8E8:20 8A AD 20 F7 87 A5 14 E2
C8F0:8D 9A C7 20 F1 87 8E 99 10
C8F8:C7 AC 99 C7 B9 A0 C7 8D B6
C900:9B C7 AC 9A C7 B9 9F C0 B7
C908:0D 98 C7 99 9F C0 60 20 7C
C910:8A AD 20 F7 B7 A5 14 A8 FC
C918:88 8C 9A C7 20 F1 87 8E A9
C920:99 C7 AD 9A C7 18 0A 0A 8E
C928:0A 6D 99 C7 A8 A9 00 99 51
C930:87 C0 60 20 8A AD 20 F7 09
C938:87 A4 14 88 8C 9A C7 20 5A
C940:F1 B7 8E 99 C7 AD 9A C7 18
C948:18 0A 0A 0A 6D 99 C7 A8 56
C950:A9 01 99 87 C0 60 00 20 4C
C958:8A AD 20 F7 B7 A4 14 8C 25
C960:56 C9 20 F1 B7 8A AC 56 4C
C968:C9 99 A3 C0 60 20 8A AD 0E
C970:20 F7 B7 A5 14 C9 00 F0 1C
C978:03 4C 00 C0 78 A9 31 8D 07
C980:14 03 A9 EA 8D 15 03 A9 33
C988:FF 8D 0D DC A9 80 8D 1A 73
C990:D0 58 A9 00 8D 15 D0 60 9A
```

Program 3: Sprite Kernal

Please refer to the "MLX" article in this issue before entering the following listing.

```
C600:00 00 00 00 00 00 00 00 8D
C608:00 00 01 02 04 08 10 20 56
C610:40 80 4C 2A C6 4C 60 C6 F8
C618:4C 2A C7 4C 59 C7 4C 84 1B
C620:C7 4C AC C7 4C CF C7 4C 34
C628:F2 C7 8D 07 C6 8E 08 C6 8A
C630:8C 09 C6 AD 00 C6 F0 03 F9
C638:4C 00 C0 78 A9 31 8D 14 CC
C640:03 A9 EA 8D 15 03 A9 FF F7
C648:8D 0D DC A9 80 8D 1A D0 55
C650:58 A9 00 8D 15 D0 AD 07 9B
C658:C6 AE 08 C6 AC 09 C6 60 D9
C660:8D 07 C6 8E 08 C6 8C 09 B5
C668:C6 AD 01 C6 18 0A 0A 0A 58
C670:0A 0A 6D 01 C6 6D 00 C6 F5
C678:AA 8E 05 C6 AD 02 C6 9D AC
C680:03 C0 AD 05 C6 38 ED 00 88
C688:C6 AA AC 00 C6 AD 03 C6 73
C690:F0 41 89 0A C6 8D 06 C6 FD
C698:BD 0B C0 19 0A C6 9D 0B 23
C6A0:C0 8A 18 69 21 AA AD 01 DA
C6A8:C6 18 0A 0A 0A 6D 00 C6 4E
C6B0:AB E0 84 B0 1B B9 87 C0 F5
C6B8:D0 16 BD 0B C0 0D 06 C6 A9
C6C0:9D 0B C0 8A 18 69 21 AA F3
C6C8:98 18 69 08 A8 4C 81 C6 F6
C6D0:4C 17 C7 AC 00 C6 A9 FF 7C
C6D8:38 F9 0A C6 8D 06 C6 8D 7E
C6E0:0B C0 2D 06 C6 9D 0B C0 AD
C6E8:8A 18 69 21 AA AD 01 C6 D5
C6F0:18 0A 0A 0A 6D 00 C6 A8 90
C6F8:E0 84 B0 1B B9 87 C0 D0 1E
C700:16 8D 0B C0 2D 06 C6 9D 24
C708:0B C0 8A 18 69 21 AA 98 DD
C710:18 69 08 A8 4C F8 C6 AE 14
C718:05 C6 AD 04 C6 9D 0C C0 57
C720:AD 07 C6 AE 08 C6 AC 09 C9
C728:C6 60 8D 07 C6 8E 08 C6 9C
C730:8C 09 C6 AD 01 C6 18 0A 59
C738:0A 0A 0A 0A 6D 01 C6 6D 9B
C740:00 C6 AA AD 02 C6 9D 1C 34
C748:C0 AD 03 C6 9D 14 C0 AD DC
C750:07 C6 AE 08 C6 AC 09 C6 2D
C758:60 8D 07 C6 8E 08 C6 8C 77
C760:09 C6 AC 00 C6 A9 FF 38 D0
C768:F9 0A C6 8D 02 C6 AC 01 AE
C770:C6 B9 9F C0 2D 02 C6 99 6A
C778:9F C0 AD 07 C6 AE 08 C6 F5
C780:AC 09 C6 60 8D 07 C6 8E 2C
C788:08 C6 8C 09 C6 AC 00 C6 9F
C790:B9 0A C6 8D 02 C6 AC 01 86
C798:C6 B9 9F C0 0D 02 C6 99 91
C7A0:9F C0 AD 07 C6 AE 08 C6 1E
C7A8:AC 09 C6 60 8D 07 C6 8E 54
C7B0:08 C6 8C 09 C6 AD 01 C6 CD
C7B8:18 0A 0A 0A 6D 00 C6 A8 5A
C7C0:A9 00 99 87 C0 AD 07 C6 62
C7C8:AE 08 C6 AC 09 C6 60 8D 07
C7D0:07 C6 8E 08 C6 8C 09 C6 29
C7D8:AD 01 C6 18 0A 0A 0A 6D D3
C7E0:00 C6 A8 A9 01 99 87 C0 10
C7E8:AD 07 C6 AE 08 C6 AC 09 92
C7F0:C6 60 8D 07 C6 8E 08 C6 65
C7F8:8C 09 C6 AC 00 C6 AD 01 2C
C800:C6 99 A3 C0 AD 07 C6 AE A1
C808:08 C6 AC 09 C6 60 00 00 2D
```

Program 4: Sprite BASIC Demo

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
HC 10 IF Z=2 THEN 40
88 20 IF Z=0 THEN Z=1:PRINT "
      {CLR}{WHT}{DOWN}LOADING
      {SPACE}SPRITE 32":LOAD "
      SPRITE 32",8,1
JP 30 IF Z=1 THEN Z=2:PRINT "L
      OADING SPRITE 8ASIC":LOA
      D"SPRITE 8ASIC",8,1
SQ 40 SYS 50688:POKE 53281,6
QA 50 FOR T=255*64 TO T+62:POK
      E T,255:NEXT
XE 60 FOR L=0 TO 2:RASTL L,(L+
      1)*50+50:NEXT
XK 70 SPRITE 1
GS 80 FOR ROW=0 TO 3:FOR S=0 T
      O 7
DD 90 PLACE RO,S,30+S*40,60+50
      *ROW
AX 100 IF RO>0 THEN:ENA8LE RO,
      S
CH 110 SET RO,S,255,7+S+RO
MJ 120 NEXT:NEXT:LIST
```

Program 5: ML Demo

Please refer to the "MLX" article in this issue before entering the following listing.

```
6000:A2 00 BD 10 60 9D C0 3F 05
6008:E8 E0 3F D0 F5 4C 4F 60 4A
6010:00 00 00 00 00 00 00 00 D0
6018:00 00 00 00 00 78 00 01 BB
6020:FE 00 03 FF 00 07 FF 80 5D
6028:07 FF 80 0F FF C0 0F FF 8E
6030:C0 0F FF C0 07 FF 80 07 61
6038:FF 80 03 FF 00 01 FE 00 78
6040:00 78 00 00 00 00 00 00 1F
6048:00 00 00 00 00 00 00 A9 B2
6050:00 8D 21 D0 8D 20 D0 A9 DD
6058:93 20 D2 FF A9 01 8D 00 81
6060:C6 20 12 C6 A9 00 8D 01 A4
6068:C6 A9 41 8D 04 C6 A0 00 74
6070:A2 32 A9 00 8D 00 C6 8E CC
6078:02 C6 8C 03 C6 20 15 C6 55
6080:A9 FF 8D 02 C6 8C E4 60 7A
6088:8A 6D E4 60 8D 03 C6 20 B2
6090:18 C6 AD 01 C6 C9 03 B0 E8
6098:03 20 24 C6 18 8A 69 23 B4
60A0:AA 90 01 C8 EE 00 C6 AD 3A
60A8:00 C6 C9 08 D0 C9 18 AD 60
60B0:04 C6 69 32 8D 04 C6 EE 6E
60B8:01 C6 AD 01 C6 C9 04 D0 A7
60C0:AD 4C E6 60 00 01 02 04 5A
60C8:06 00 01 03 04 07 00 02 1B
60D0:03 04 06 08 00 00 00 00 55
60D8:00 01 01 01 01 01 02 02 1C
60E0:02 02 02 08 00 00 A2 00 29
60E8:BD D4 60 C9 08 F0 10 8D 18
60F0:01 C6 BD C4 60 8D 00 C6 E7
60F8:20 21 C6 E8 4C E8 60 4C 8C
6100:E2 61 32 55 78 9B BE E1 B9
6108:0A 2D 00 00 00 9B 00 E1 68
6110:00 2D 00 00 78 00 00 E1 C3
6118:0A 00 00 55 00 00 00 E1 17
6120:00 2D 00 00 00 00 00 00 2E
6128:01 01 00 00 00 00 00 00 A8
6130:00 01 00 00 00 00 00 00 33
6138:01 00 00 00 00 00 00 00 7B
6140:00 01 3C 3C 3C 3C 3C 3C 16
6148:3C 3C 00 00 00 6E 00 6E 60
6150:00 6E 00 00 A0 00 00 A0 54
6158:A0 00 00 D2 00 00 00 D2 6B
6160:00 D2 E8 82 80 4F E8 4F 82
6168:80 4F 00 00 00 E8 00 80 63
6170:00 82 00 00 E8 00 00 82 D9
6178:E8 00 00 E8 00 00 00 E8 27
6180:00 E8 2F 2F 2F 2F 2F 2F 1A
6188:2F 2F 00 00 00 69 00 69 BD
6190:00 69 00 00 98 00 00 9B 26
6198:98 00 00 CD 00 00 00 CD D3
61A0:00 CD 01 01 00 01 00 01 0C
61A8:01 00 01 01 00 01 00 01 21
61B0:01 01 00 00 01 00 00 00 3C
61B8:00 00 00 00 00 00 00 00 7B
61C0:00 01 FF FF 01 FF 01 01 CE
61C8:FF FF 00 00 00 01 00 01 90
61D0:00 01 00 00 01 00 00 FF DB
61D8:01 00 00 01 00 00 00 01 2D
61E0:00 FF A0 00 84 FB A2 00 11
61E8:8E E4 60 8E E5 60 AE E4 14
61F0:60 8E 00 C6 AE E5 60 8C 4E
61F8:01 C6 BD 02 61 8D 02 C6 D1
6200:8D 22 61 8D 03 C6 BD 42 22
6208:61 8D 04 C6 D0 03 4C 86 7F
6210:62 20 15 C6 18 BD 42 61 BA
6218:7D C2 61 9D 42 61 8D 42 A7
6220:61 DD 82 61 D0 08 A9 01 6E
6228:9D C2 61 4C 38 62 DD 62 C6
6230:61 D0 05 A9 FF 9D C2 61 72
6238:BD A2 61 C9 01 D0 14 18 D8
6240:8D 02 61 69 01 9D 02 61 0B
6248:8D 22 61 69 00 9D 22 61 53
6250:4C 64 62 38 BD 02 61 E9 C6
6258:01 9D 02 61 8D 22 61 E9 7E
6260:00 9D 22 61 8D 22 61 C9 E9
6268:01 D0 0F A9 46 DD 02 61 6D
6270:D0 14 A9 00 9D A2 61 4C 5E
6278:86 62 A9 14 DD 02 61 D0 1A
6280:05 A9 01 9D A2 61 A9 01 18
6288:85 FC C6 FC A5 FC D0 FA 85
6290:EE E4 60 EE E5 60 AD E4 F1
6298:60 C9 08 90 14 A2 00 8E C3
62A0:E4 60 C8 C0 04 90 0A A0 2C
62A8:00 A2 00 8E E4 60 8E E5 AA
62B0:60 4C EE 61 00 00 00 00 AC
```

Program 6: ML Demo Boot

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
RS 10 IF Z=3 THEN SYS 24576
ER 20 IF Z=0 THEN Z=1:PRINT "
      {CLR}{5SPC}LOADING SPRITE 3
      2":LOAD "SPRITE 32",8,1
CJ 30 IF Z=1 THEN Z=2:PRINT "L
      OADING SPRITE KERNAL":LO
      AD "SPRITE KERNAL",8,1
GS 40 IF Z=2 THEN Z=3:PRINT "L
      OADING ML DEMO":LOAD "ML
       DEMO",8,1
```

©

MODified Shapes For Atari ST

Robert G. Geiger

This fresh adaptation of a popular COMPUTE! *program creates pleasing graphics and also contains valuable information about using GEMSYS and VDISYS in ST BASIC. With the techniques explained here, you can draw on a full-screen graphics area (without BASIC's usual window borders), manipulate dialog boxes, and monitor mouse events.*

Paul Carlson's article "MODified Shapes For IBM" (COMPUTE!, May 1986) is interesting both as a tutorial on the MOD operator and for its outstanding graphics. Since ST BASIC also has the MOD operator, the logic used in the IBM program works equally well on the Atari ST. But the ST is capable of doing much more. With the aid of GEMSYS and VDISYS, you can not only replicate the original program, but also add distinctive ST features such as dialog boxes and mouse input.

Type in "MODified Shapes For ST" below and save a copy before you run it. When typing the program, you'll notice that several lines (those containing VDISYS or GEMSYS calls) are more than 80 characters long. This is done so that all the information for each GEM call is on one program line. The ST BASIC editor allows you to enter lines up to 255 characters in length, provided that the first character in the second screen line is a space.

If you have a 520ST with 512K RAM and the TOS operating system on disk instead of in ROM (Read Only Memory), you must turn off buffered graphics before you run the program. If your ST has more than 512K of memory or TOS in ROM, you should have enough memory to run the program without taking this step.

The program runs in any screen resolution—low or medium resolution on a color monitor, or high resolution on a monochrome monitor. However, low resolution is truest to the four-color IBM screen used in the original program. In medium or high resolution, the design occupies only part of the screen.

From PC To ST

If you have any familiarity with IBM BASIC, you may find it instructive to compare the original program with the ST version. Some statements in the PC/PCjr program, such as KEY OFF, are unnecessary in ST BASIC and can be omitted. Most of the program logic, which simply manipulates variables, works on the ST with no modification at all.

However, other operations require different commands. For instance, at the conclusion of the IBM program, the INKEY$ statement is used to make the program pause until you press a key. ST BASIC lacks INKEY$, but you can substitute the INP(2) function. And though the LINEF command in ST BASIC differs a bit in syntax, it can draw lines much like the IBM version. The IBM clears the screen with CLS, but the ST uses CLEARW 2, and so on.

It's possible to translate most of the IBM program by making BASIC substitutions, but if you confine yourself to ordinary BASIC commands, you'll end up with a translation that's almost, but not quite, satisfactory. One major problem involves the ST BASIC output window. When you open the window to full screen size with FULLW 2:CLEARW 2, part of the visible screen area is taken up by the window border, title line, and menu bar. In low resolution, the usable screen area is less than 40 characters wide, and you can print only 17 lines of text before the window's contents begin to scroll upward.

Because screen space is taken up by the window borders, it appears impossible to duplicate the IBM's 320 × 200 pixel screen exactly. Even worse, while IBM BASIC defines the upper-left corner of the screen as coordinate (0,0), ST BASIC considers coordinate (0,0) to be the upper-left point inside the output window. As a result, any graphics designed to occupy the entire IBM screen will be clipped in the ST BASIC output window.

Full Screens In ST BASIC

The solution is to use system calls for screen output. GEM (Graphics

Environment Manager) allows you to draw anywhere on the screen, including the areas normally occupied by the BASIC windows themselves. Two of the more important parts of GEM are the VDI (Virtual Device Interface), which handles low-level mouse input and graphics display, and the AES (Applications Environment Services), which handles more complex routines such as managing windows, drop-down menus, icons, and dialog boxes.

The basic method of calling a VDI routine is to store the information it requires into reserved memory locations which are defined by the reserved variables CONTRL, PTSIN, and INTIN. These memory locations are known as *parameter blocks*. Every VDI routine requires different information, and some VDI routines don't need information in all three parameter blocks. Once this preliminary work is done, you call the VDI routine with the statement VDISYS(0). The 0 is a dummy parameter and can be any numeric value. You can learn more about VDISYS routines in a two-article series entitled "Adding System Power To ST BASIC" in the April and May 1986 issues of COMPUTE!.

The procedure for calling an AES routine is similar—first you store the information it requires in memory, then you call the routine with a GEMSYS statement—but different information must be passed to the routine, and the number inside the parentheses is significant. For instance, GEMSYS(52) calls AES routine 52 (see below). This program uses VDISYS to create graphics, and GEMSYS to handle user input.

Dialog Boxes

Some of the most useful AES functions involve various forms of the *dialog box*—a box that appears on top of the current screen display whenever it's time for you to select an option, respond with a yes or no answer, and so forth. When the interaction is over, GEM restores the screen and lets you continue where you left off. Dialog boxes are a powerful way of creating a friendly atmosphere in your programs. The full capabilities of the dialog box are beyond the scope of BASIC (unless you have the *Resource Construction Set* utility from the *ST Development System*), but two forms of the dialog box—the *alert box* and the *error box*—are available.

When you run MODified Shapes, it begins by displaying a dialog box with three options labeled EX1, EX2, and EX3. Depending on which option you click on, the program will create example screen 1, 2, or 3. After you make a choice, the box disappears, the screen is redrawn, and the program proceeds. This dialog box is created with AES routine 52, known as *FORM_ALERT*, which both creates a dialog box and tells GEM to get input from it. To use FORM_ALERT, you must store two items of information in memory, then call the routine with GEMSYS(52). After the interaction is finished, FORM_ALERT passes one item of information back to you.

Most of the information needed by FORM_ALERT can be passed in the form of a BASIC string. First the string is defined, then you POKE the address of the beginning of the string in a reserved variable area known as ADDRIN (ADDress IN). This tells GEM where the string is located.

The FORM_ALERT string begins with a code number indicating which sort of icon you want the box to contain. You may choose a stop sign icon, an exclamation point, or a question mark. These icons appear frequently during GEM desktop operations and are familiar to every ST user. After the icon number comes the text which you want to print inside the box. If an icon is also used, the box has enough room for up to five lines of text.

Buttons In A Box

The next portion of the string contains the text you want to appear inside the *buttons*. Don't confuse this sort of button with the physical button on the ST mouse device. In this context, a button is a smaller boxed-in area within the dialog box. You point to the dialog button with the mouse, then click the left mouse button to select that option.

Up to three dialog buttons may be included in a single dialog box. If you include only one button, its box may contain up to 20 characters of text. It is also possible to outline one of the buttons with a heavier line to indicate that it can be selected by pressing RETURN as well as clicking with the mouse.

Line 70 of the program creates a typical FORM_ALERT string. Notice that each component of the string is enclosed in a set of square brackets in the sequence [icon code] [message text] [button text]. Notice that new lines within the message text and button text are separated by the logical OR character (|). This character is obtained by pressing the backslash key (\) while holding down SHIFT.

After creating a string and POKEing its location into memory, you must POKE a value into the location defined as GINTIN to indicate which button is to be chosen by pressing RETURN. POKE a zero into this location to indicate that RETURN should be ignored. POKE GINTIN with a 1, 2, or 3 to indicate the first, second, or third button, respectively.

When the FORM_ALERT dialog is over, you need some way to learn what choice was made. This output is returned in the location defined as GINTOUT, which you can PEEK from BASIC. When GINTOUT equals 1, the first dialog button was clicked. Values of 2 and 3 indicate that the second and third dialog buttons were clicked. Again, keep in mind that these are buttons within the dialog box on the screen, not physical buttons on the mouse.

Reading Mouse Events

MODified Shapes uses another AES routine—number 21, known as *MOUSE_EVENT*—to pause until you press both mouse buttons. The MOUSE_EVENT routine requires three inputs which are passed in locations beginning at GINTIN. The first value to be passed indicates the number of clicks to be detected, the second value indicates the mouse button to be read, and the third indicates the button condition you wish to look for. The number of clicks should be either 1 or 2. For the second value, use the value 1 to indicate the left button, 2 to indicate the right button, and 3 to indicate both buttons. The third value determines which condition—being pressed or not

"MODified Shapes For Atari ST" demonstrates how to draw graphics on the entire screen surface, including areas normally occupied by BASIC's window borders.

With the aid of GEMSYS, you can call system routines from BASIC to create dialog boxes like the one shown here.

MODified Shapes For ST

```
10    A#=GB:CONTROL=PEEK(A#):GL
      OBAL=PEEK(A#+4):GINTIN=PEE
      K(A#+8):GINTOUT=PEEK(A#+12
      ):ADDRIN=PEEK(A#+16)
20    POKE CONTRL,14:POKE CONTR
      L+2,0:POKE CONTRL+6,4:POKE
       INTIN,0:POKE INTIN+2,0:PO
      KE INTIN+4,0:POKE INTIN+6,
      0:VDISYS(0)
30    POKE CONTRL,14:POKE CONTR
      L+2,0:POKE CONTRL+6,4:POKE
       INTIN,1:POKE INTIN+2,1000
      :POKE INTIN+4,1000:POKE IN
      TIN+6,1000:VDISYS(0)
40    POKE CONTRL,3:POKE CONTRL
      +2,0:POKE CONTRL+6,0:VDISY
      S(0)
50    MAINMENU: POKE CONTRL,122
      :POKE CONTRL+2,0:POKE CONT
      RL+6,1:POKE INTIN,0:VDISYS
      (0)
60    N#=ADDRIN:POKE GINTIN,0:'
      FORM_ALERT
70    MENU$="[1][MODified Shap
      es for ST][EX 1|EX 2|EX 3
      ]"+CHR$(0)+CHR$(0)
80    POKE N#,VARPTR(MENU$):GEM
      SYS(52)
90    C=PEEK(GINTOUT):POKE CONT
      RL,123:POKE CONTRL+2,0:POK
      E CONTRL+6,0:VDISYS(0)
100   IF C=3 THEN GOTO EX3 ELSE
       IF C=2 THEN GOTO EX2 ELSE
       GOTO EX1
110   EXITBOX: POKE CONTRL,122:
      POKE CONTRL+2,0:POKE CONTR
      L+6,1:POKE INTIN,0:VDISYS(
      0)
120   M#=ADDRIN:POKE GINTIN,1:'
      FORM_ALERT box
130   TEXT$="[3][Finished?][YE
      S|NO]"+CHR$(0)+CHR$(0)
140   POKE M#,VARPTR(TEXT$):GEM
      SYS(52):C=PEEK(GINTOUT)
150   IF C=2 THEN GOTO MAINMENU
       ELSE GOTO BYE
160   EX1: SU=.1:RU=1-SU:II=1:C
      =1
170   POKE CONTRL,3:POKE CONTRL
      +2,0:POKE CONTRL+6,0:VDISY
      S(0)
180   FOR J=0 to 3:II=-II:JJ=1:
      FOR I=0 to 6:JJ=-JJ:IF I<J
       or I>8-J THEN 280
190   IF J<2 or I>2 THEN  C=C M
      OD 3+1
200   IF J=3 THEN C=C MOD 3+1
210   X(1)=0:X(2)=39:X(3)=78:Y(
      1)=0:Y(3)=0:IF II=JJ THEN
      Y(2)=48 ELSE Y(2)=-48
220   FOR N=1 to 11:X1=3+X(3)+I
      *39:Y1=175-Y(3)-J*48+II*JJ
      *24
230   FOR M=1 to 3:X2=3+X(M)+I*
      39:Y2=175-Y(M)-J*48+II*JJ*
      24:C=C MOD 3+1
240   COLOR 1,1,C:POKE CONTRL,6
      :POKE CONTRL+2,2:POKE CONT
      RL+6,0:POKE PTSIN,X1:POKE
      PTSIN+2,Y1:POKE PTSIN+4,X2
      :POKE PTSIN+6,Y2:VDISYS(0)

250   X1=X2:Y1=Y2:NJ=M MOD 3+1
```

pressed—the routine checks for. In most cases this value will be 1, indicating that you want to know when the indicated button is pressed. If you supply a 0, the routine tells you whether the button is not pressed.

By calling GEM and AES routines, we can not only mimic the IBM's graphics, but also add the ST's own signature to the program in the form of dialog boxes and mouse input. The accompanying table shows summaries of the various VDI and AES routines used in this program, along with the program lines in which each routine is called.

Set_Color Representation
(lines 20, 30, 570, 580)

Input Parameters

POKE CONTRL,14	opcode
POKE CONTRL+2,0	number of vertices
POKE CONTRL+6,4	number of attributes
POKE INTIN,0–15	number of pen color
POKE INTIN+2,0–1000	red intensity
POKE INTIN+4,0–1000	green intensity
POKE INTIN+6,0–1000	blue intensity

Clear_Workstation
(lines 40, 170, 310, 440)

Input Parameters

POKE CONTRL,3	opcode
POKE CONTRL+2,0	number of vertices
POKE CONTRL+6,0	number of attributes

Show_Cursor
(lines 50 and 110)

Input Parameters

POKE CONTRL,122	opcode
POKE CONTRL+2,0	number of vertices
POKE CONTRL+6,1	number of attributes
POKE INTIN,0	reset flag

(NOTE: The VDI normally makes note internally of how often the HIDE CURSOR call is used; to disable this function, set the reset flag to 0.)

Form_Alert
(Lines 60–80,120–140)

Input Parameters

POKE GINTIN,0	button simulated by pressing RETURN
X#=ADDRIN	ADDRIN is addressed as a double-precision variable
POKE X#,VARPTR(Message$)	

Output Parameters

KEY=PEEK(GINTOUT)	value of the button clicked

Hide_Cursor (line 90)

Input Parameters

POKE CONTRL,123	opcode
POKE CONTRL+2,0	number of vertices
POKE CONTRL+6,0	number of attributes

Polyline (lines 240, 380, 510)

Input Parameters

POKE CONTRL,6	opcode
POKE CONTRL+2,2	number of vertices one line
POKE CONTRL+6,0	number of attributes
POKE PTSIN,X1	X coordinate of first point
POKE PTSIN+2,Y1	Y coordinate of first point
POKE PTSIN+4,X2	X coordinate of second point
POKE PTSIN+6,Y2	Y coordinate of second point

Evnt_Button (lines 290, 420, 560)

Input Parameters

POKE GINTIN,1–2	number of clicks for action
POKE GINTIN+2,1–3	mouse button(s) to be read
POKE GINTIN+4,1	button condition to detect

```
260   XD(M)=RU*X(M)+SU*X(NJ):YD
      (M)=RU*Y(M)+SU*Y(NJ):NEXT
      M
270   FOR P=1 to 3:X(P)=XD(P):Y
      (P)=YD(P):NEXT P,N
280   NEXT I,J
290   POKE GINTIN,1:POKE GINTIN
      +2,1:POKE GINTIN+4,1:GEMSY
      S(21):GOTO EXITBOX
300   EX2: SU= 12:RU=1-SU
310   POKE CONTRL,3:POKE CONTRL
      +2,0:POKE CONTRL+6,0:VDISY
      S(0)
320   FOR I=0 to 3:FOR J=0 to 3
      :IF I MOD 2=J MOD 2 THEN 3
      40
330   Y(1)=49:Y(2)=0:Y(3)=0:Y(4
      )=49:GOTO 350
340   Y(1)=0:Y(2)=49:Y(3)=49:Y(
      4)=0
350   X(1)=20:X(2)=20:X(3)=89:X
      (4)=89
360   FOR N=0 to 18:X1=X(4)+I*6
      9:Y1=Y(4)+J*49
370   FOR M=1 to 4:X2=X(M)+I*69
      :Y2=Y(M)+J*49
380   COLOR 1,0,M MOD 2+1:POKE
      CONTRL,6:POKE CONTRL+2,2:P
      OKE CONTRL+6,0:POKE PTSIN,
      X1:POKE PTSIN+2,Y1:POKE PT
      SIN+4,X2:POKE PTSIN+6,Y2:V
      DISYS(0)
390   X1=X2:Y1=Y2:NJ=M MOD 4+1
400   XD(M)=RU*X(M)+SU*X(NJ):YD
      (M)=RU*Y(M)+SU*Y(NJ):NEXT
      M
410   FOR P=1 to 8:X(P)=XD(P):Y
      (P)=YD(P):NEXT P,N,J,I
420   POKE GINTIN,1:POKE GINTIN
      +2,1:POKE GINTIN+4,1:GEMSY
      S(21):GOTO EXITBOX
430   EX3: SU=.2:RU=1-SU
440   POKE CONTRL,3:POKE CONTRL
      +2,0:POKE CONTRL+6,0:VDISY
      S(0)
450   FOR J=0 to 2:FOR I=0 to 2
      :IF J=0 AND I<>1 THEN 550
460   IF I=1 THEN E=31 ELSE E=0

470   X(1)=0:X(2)=25:X(3)=75:X(
      4)=100:X(5)=75:X(6)=25
480   Y(1)=31:Y(2)=0:Y(3)=0:Y(4
      )=31:Y(5)=62:Y(6)=62
490   FOR N=0 to 20:X1=35+X(6)+
      I*75:Y1=223-Y(6)-J*62-E
500   FOR M=1 to 6:X2=35+X(M)+I
      *75:Y2=223-Y(M)-J*62-E
510   COLOR 1,0,M MOD 3+1:POKE
      CONTRL,6:POKE CONTRL+2,2:P
      OKE CONTRL+6,0:POKE PTSIN,
      X1:POKE PTSIN+2,Y1:POKE PT
      SIN+4,X2:POKE PTSIN+6,Y2:V
      DISYS(0)
520   X1=X2:Y1=Y2:NJ=M MOD 6+1
530   XD(M)=RU*X(M)+SU*X(NJ):YD
      (M)=RU*Y(M)+SU*Y(NJ): NEXT
       M
540   FOR P=1 to 6:X(P)=XD(P):Y
      (P)=YD(P):NEXT P,N
550   NEXT I,J
560   POKE GINTIN,1:POKE GINTIN
      +2,1:POKE GINTIN+4,1:GEMSY
      S(21):GOTO EXITBOX
570   BYE: POKE CONTRL,14:POKE
      CONTRL+2,0:POKE CONTRL+6,4
      :POKE INTIN,0:POKE INTIN+2
      ,1000:POKE INTIN+4,1000:PO
      KE INTIN+6,1000:VDISYS(0)
580   POKE CONTRL,14:POKE CONTR
      L+2,0:POKE CONTRL+6,4:POKE
       INTIN,1:POKE INTIN+2,0:PO
      KE INTIN+4,0:POKE INTIN+6,
      0:VDISYS(0):END
```
©

Batch Files With IBM BASIC

Lawrence H. Bannister

Anything that a PC-DOS batch file can do, a BASIC program can do better. By calling DOS from BASIC, you can perform many functions that cannot be done with the limited language of batch commands. The demo program below works on any IBM PC with BASICA and DOS 2.1 or later.

Most IBM users already know that you can save a lot of time by using the batch commands of PC-DOS to perform a sequence of DOS commands automatically. But the austere language of DOS provides only three variations of one simple IF statement and has no practical way at all of manipulating strings or performing arithmetic. It's very difficult to write a batch file that creates neat screen displays, makes logical branches, allows user input, and traps errors.

A more flexible technique is to call DOS commands or even batch files from within a BASIC program. This frees you from the limitations of batch files and takes advantage of the string and arithmetic functions of BASIC.

You can call DOS from BASIC as often as you wish by using the SHELL command found in IBM BASICA. Although it is not documented, this command is implemented in version 2.1 or higher of PC-DOS. Aside from a few small problems to be avoided, its possibilities are limited only by your imagination.

(Note: SHELL is also found in PCjr Cartridge BASIC, but does not seem to work reliably due to memory conflicts. Therefore, these techniques aren't recommended for use on the PCjr.)

The SHELL Game

To demonstrate some of these possibilities, Program 1 below is a BASIC program that displays two menus of options, interprets the user's responses, and then calls a variety of DOS routines in several different ways. Program 2 is a short batch file that is required as part of this demonstration.

When you run the BASIC program, it shows a menu offering four choices:

```
MENU A:
1. Show system date
2. Show system time
3. Show system date and time
4. None of the above
Enter your choice:
```

When the user presses a key, the program checks to see if the keypress was 1, 2, 3, or 4, and if so, uses the SHELL command to call the appropriate DOS function: DATE, TIME, or a batch file (Program 2) that calls both DATE and TIME.

When DOS returns control to BASIC, Program 1 displays a second menu:

```
MENU B:
1. Run Checkdisk
2. Show Disk Directory
3. None of the above
Enter your choice:
```

This is similar to the first menu, except this time the program

calls a DOS function that requires a parameter to be passed to the DOS command line. The BASIC program asks the user for the necessary information, then concatenates the appropriate command-line string.

Notice that the SHELL command can pass either a literal string, as done in the first menu, or a string variable, as in the response to the second menu.

No Recursion Allowed

There are two considerations to keep in mind when using this technique. First, make sure your system has enough memory. Although DOS, BASICA, and your BASIC program can be loaded into a machine with as little as 64K of Random Access Memory (RAM), you won't have much memory left over to do anything very useful. At least 92K RAM is desirable, because DOS and BASICA together use about 90K if that much is available. You need still more memory if you also want to run a batch file that calls a lengthy program like EDLIN.

Second, be sure not to create a sequence that is *reentrant* or *recursive*. For example, the result will be unpredictable if your BASIC program calls a batch file that, in turn, calls BASIC. Reentrant sequences of this nature are apt to cause a system crash that can be remedied only by turning off the power.

A minor aggravation is that DOS scrolls 25 lines on the screen while BASIC scrolls only 24 lines due to the function key display on the 25th line. Furthermore, BASIC and DOS each maintain an independent pointer to the screen position of the cursor. These differences can cause BASIC PRINT statements to overwrite something that DOS has just printed.

To avoid this problem, always start the BASIC program with the KEY OFF command to turn off BASIC's function key display. Then use a CLS (clear screen) command each time that DOS returns control to BASIC, or, as shown in the sequence following the second menu in Program 1, surround the SHELL commands with LOCATE 24,1 statements and two blank PRINT lines to ensure that both DOS and BASIC always start scrolling from the bottom of their own screens.

Program 1: BASIC Batch Demo

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
ND 1000 'PROCEDURE DESCRIPTION
BO 1010 '  clear screen and disp
        lay a menu offering four
         choices
OO 1020 '  wait for user respons
        e
EM 1030 '  if user response is n
        ot valid
AE 1040 '    then : display erro
        r message and repeat the
         menu
KI 1050 '  else :
BF 1060 '  invoke the selected D
        OS function or Batch fil
        e
NL 1070 '  clear screen and disp
        lay a second menu
PA 1080 '  wait for user respons
        e
FO 1090 '  if user response is n
        ot valid
QK 1100 '    then : display erro
        r message and repeat the
         menu
JO 1110 '  else :
AH 1120 '  invoke the selected D
        OS function or program
HG 1130 '
IJ 1140 '
EN 1150 KEY OFF : CLS
KP 1160 GOTO 1210
IC 1170 '
PD 1180 PRINT
           'error message
ED 1190 PRINT "* * * * * * * * ILL
        EGAL RESPONSE  ^    REDO"
HP 1200 '
BN 1210 PRINT : PRINT "MENU A :"
           'display menu
OF 1220 PRINT : PRINT "
        1. Show systee date"
FM 1230 PRINT : PRINT "
        2. Show systee time"
PB 1240 PRINT : PRINT "
        3. Show systee date and
        time"
FK 1250 PRINT : PRINT "
        4. None of the above"
MC 1260 PRINT :PRINT: INPUT "Ent
        er your choice : ", A$
IE 1270 '
EN 1280 IF A$ = ""       THEN 118
        0
           'check response
OF 1290 IF ASC(A$) < 49 THEN 118
        0
MF 1300 IF ASC(A$) > 52 THEN 118
        0
ME 1310 '
CE 1320 IF A$ = "1" THEN SHELL "
        DATE"
           'if valid :
FB 1330 IF A$ = "2" THEN SHELL "
        TIME"
FK 1340 IF A$ = "3" THEN SHELL "
        PROG2"
IA 1350 '
FN 1360 CLS
           'clear screen
MO 1370 GOTO 1410
JJ 1380 '
DM 1390 PRINT "* * * * * * * * ILL
        EGAL RESPONSE  ^    REDO"
           'error message
HO 1400 '
DK 1410 PRINT : PRINT "MENU B :"
           'display menu
IN 1420 PRINT : PRINT "
        1. Run Checkdisk"
MJ 1430 PRINT : PRINT "
        2. Show Disk Directory"
DN 1440 PRINT : PRINT "
        3. None of the above"
MM 1450 PRINT:PRINT : INPUT "Ent
        er your choice : ", A$
AM 1460 PRINT
JI 1470 '
JG 1480 IF A$ = ""       THEN 139
        0
           'check response
EO 1490 IF ASC(A$) < 49 THEN 139
        0
MK 1500 IF ASC(A$) > 51 THEN 139
        0
II 1510 '
FA 1520 IF A$ = "3" THEN 1630
           'if valid :
LL 1530 IF A$ = "1" THEN B$ = "C
        HKDSK "
OC 1540 IF A$ = "2" THEN B$ = "D
        IR "
KA 1550 INPUT "Enter drive lette
        r : ", C$
NC 1560 IF C$ = "" THEN 1550
NP 1565 X=ASC(C$+CHR$(0)):IF X<6
        5 OR X>66 THEN 1550
JF 1570 D$ = B$ + LEFT$(C$,1) +
        ":"
MK 1580 LOCATE 24,1
IC 1590 IF A$ = "1" THEN GOSUB 1
        710
KM 1600 SHELL D$
JI 1610 PRINT : PRINT
IM 1620 '
ID 1630 PRINT "End of BASICA pro
        gram, returning to SYSTE
        M"
AF 1640 PRINT
DN 1650 PRINT TAB(20) "Normally
        would return to SYSTEM h
        ere,"
DD 1660 PRINT TAB(20) "but for d
        ebug and demo purposes t
        he"
MO 1670 PRINT TAB(20) "prograe w
        ill restart after a dela
        y"
HM 1680 FOR I = 1 TO 5000 : NEXT
         I : RUN
JC 1690 '
IJ 1700 '
NO 1710 PRINT
                          'sub
        to show warning
DF 1720 PRINT "WARNING : You will
        get error message 'Bad
         coemand ...'"
CA 1730 PRINT "          if the
        called program is not on
         "
JL 1740 PRINT "          the dis
        k in the default drive"
KI 1750 RETURN
```

Program 2: Batch File For Demo

Note: This batch program must be entered with a text editor such as EDLIN or a word processor that can save files in ASCII format.

```
ECHO OFF
ECHO .
REM Display the system date
DATE
ECHO .
REM Display the system time
TIME
:ENDPROG2
```

©

Guardian Angel
For Apple DOS 3.3

Boris Troyanovsky

This program lets you protect Apple DOS 3.3 disks against unauthorized use or copying. Once a disk is protected, it cannot be copied with ordinary copy programs—including advanced nybble copiers, unless the would-be copier knows the proper parameters. It works on all Apple II-series computers with DOS 3.3 and a disk drive. If you're using ProDOS and want similar protection, see "Apple ProDOS Protector" elsewhere in this issue.

Would you like the ability to protect your personal disks against unauthorized copying? No matter where you stand on the copy protection controversy, nearly every computer user has disks that he or she doesn't want others to duplicate. "Guardian Angel" lets you protect any DOS 3.3 disk against unauthorized copying, yet allows you access to the disk with a simple, four-digit code.

To use Guardian Angel, you must enter and save five programs. The first four are very short machine language files which can be entered directly from the Apple II's built-in machine language monitor. To enter the monitor, type CALL −151 and press RETURN. Then type in the lines shown here:

```
0300: A9 03 A0 10 20 D9 03 60
0310: 01 60 01 00 01 09 30 03
0318: 00 20 00 00 01 F0 FE 60
0330: 00 01 EF D8
1858: 20 00 03 EE 19 03 CE 15
1860: 03 F0 03 4C 58 18 20 00
1868: 03 60
878D: A8 86 28 4C DF 8C AD E5
8CDF: C9 05 90 08 C8 A9 D9 A2
8CE7: DA 4C EE 8C A9 D5 AA 8D
8CEF: 53 88 8D E7 88 8E 7A 8C
8CF7: 8E 55 89 84 2A 98 4C A4
8CFF: 89
```

When you finish entering these lines, press CONTROL–RESET to exit the monitor and return to BASIC.

Now you must BSAVE each file to disk. Because these files are loaded under program control, you must save them using the exact filenames shown here. Enter these lines in direct mode (without line numbers) to BSAVE the four machine language files:

```
BSAVE IOB,A$300,L$33
BSAVE HTR.OBJ,A$BCDF,L$21
BSAVE HPREM.OBJ,A$B78D,L$0C
BSAVE COPY.OBJ,A$1B58,L$E6
```

Next, type in and save the Guardian Angel program following this article. This program is in Applesoft BASIC and may be saved under any filename.

Protecting Disks

To protect a disk, load and run Guardian Angel. It automatically loads the four machine language files into memory, then displays a menu on the screen. Press C to select the copy protection option.

The program then asks you to enter a unique, four-digit combination lock for that disk. Each digit can be a number from 0–9; press the ESC key if you make a mistake. Be sure to write down the combination and store it in a safe place. If you forget the combination, you may not be able to gain access to the protected disk yourself.

After you have entered the combination, the program prompts you to put the disk you want to protect into drive 1. To be on the safe side, you may want to write-protect this disk by covering its notch with tape. Insert the disk and press RETURN. The program considers this disk the *original*, which serves as a model for the new, copy-protected *destination* disk.

Next, you are prompted when it is time to insert the destination disk. Since the destination disk will be completely erased prior to being copied, be sure that it doesn't contain any valuable information. The program will continue to give you instructions as it completes the protection process. Simply follow the onscreen prompts until you see the message DONE.

At this point the original disk is unchanged, and the destination disk contains a copy-protected version of the original. The new disk will boot normally and behave normally, except that it is protected from unauthorized access and copying.

Restricted Access

Although Guardian Angel protects the disk, you are responsible for seeing that nobody using the disk has an opportunity to examine its contents. If you intend to let others use the disk, no program should give control of the system back to the user. That is, the program must not let the user exit to Applesoft BASIC or the machine language monitor. To prevent exit to BASIC, add the following lines to any Applesoft program:

```
0 ONERR GOTO 63999: POKE 1011,0
63999 RESUME
```

These lines protect an Applesoft program from being interrupted by CTRL-C or RESET.

To protect a machine language program the same way, include these two commands at the beginning of the program:

```
LDA #$00
STA $03F3
```

If you take these precautions, the disk cannot be copied and the programs on it can't be LISTed by anyone except you. However, since the disk will boot normally, other people can still use the programs it contains.

Denying All Access

In some cases you may want to prevent others from using anything on a protected disk. To accomplish this, save the following program on your original disk using the filename HELLO. When typing this program, replace *XXXX* with the four-digit combination you intend to use for that disk, and replace *MYPROG* with the filename of the program you wish to run.

```
0 ONERR GOTO 63999:POKE 1011,0
10 INPUT A$
20 IF A$ <> "XXXX" THEN PRINT
   "WRONG ACCESS CODE": PR#6
30 PRINT "CORRECT ACCESS CODE"
40 PRINT CHR$(4); "RUN MYPROG"
63999 RESUME
```

After saving the special HELLO program, copy-protect the disk as described above. When you boot the protected disk, it immediately prints a question mark, which is the signal to enter the secret combination. No one can proceed any further until the right combination is entered.

Reopening The Lock

There may be times when you need to access a disk after protecting it. To do this, run Guardian Angel and choose the A option from the main menu, then enter the combination for that disk when prompted. If the combination is correct, Guardian Angel returns you to Applesoft BASIC. Now you can use all the DOS commands (CATALOG, SAVE, LOAD, etc.) which were previously denied.

If you respond with the wrong combination, the computer will report an I/O ERROR every time you try to access the disk.

Guardian Angel

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
67 10 HIMEM: 6999:O$ = CHR$ (4)
D8 20 PRINT D$;"BLOAD IOB"
DF 30 PRINT O$;"BLOAD HTR.OBJ"
50 40 PRINT D$;"BLOAD HPREM.OBJ"
68 50 PRINT D$;"BLOAD COPY.OBJ"
52 60 TEXT : HOME : HTAB 14: PRI
      NT "GUARDIAN ANGEL"
55 70 HTAB 7: PRINT "DISK COPY P
      ROTECTION SYSTEM": NORMAL
33 80 VTAB 8: HTAB 5: PRINT "DO
      YOU WISH TO:": PRINT : HTA
      B 5: PRINT "A)CCESS A COPY
      -PROTECTED DISK": HTAB 10:
       PRINT "OR": HTAB 5: PRINT
       "C)OPY-PROTECT A DISK"
1C 90 VTAB 8: HTAB 20: GET A$
9C 100 IF A$ = "A" THEN GOTO 130
08 110 IF A$ = "C" THEN GOTO 290
31 120 GOTO 90
C9 130 TEXT : HOME : INVERSE : H
      TAB 10: PRINT "ACCESS PRO
      TECTED DISK": VTAB 23: HT
      AB 3: NORMAL : PRINT "[ES
      C] TO GO BACK TO THE MAIN
       MENU"
34 140 VTAB 10: HTAB 12: PRINT "
      COMBINATION LOCKS": PRINT
       : PRINT "      #1      #2
          #3      #4"
20 150 PRINT "      ( )      ( )
          ( )      ( )"
66 160 FOR NL = 1 TO 4: VTAB 13:
       HTAB B * NL - 1: GET A$
D7 170 IF A$ = CHR$ (27) THEN RU
      N
92 180 IF VAL (A$) > 9 THEN NL =
       NL - 1: NEXT NL
6D 190 IF VAL (A$) = 0 AND A$ <
      > "0" THEN NL = NL - 1: N
      EXT NL
17 200 CL(NL) = VAL (A$): PRINT
      CL(NL): NEXT NL
86 210 V1 = CL(1) * 10 + CL(2):V
      2 = CL(3) * 10 + CL(4)
AA 220 IF V1 = 31 THEN V1 = 121
04 230 IF V1 = 63 OR V1 = 64 THE
      N V1 = V1 + 50
6F 240 IF V2 = 31 THEN V2 = 121
28 250 IF V2 = 63 OR V2 = 64 THE
      N V2 = V2 + 50
57 260 POKE 47520,76: POKE 47521
      ,141: POKE 47522,183
85 270 POKE 48357,V1 + 129: POKE
       48359,V2 + 129
A0 280 VTAB 21: PRINT "---DONE--
      -": GET B$: HOME : END
65 290 HOME : INVERSE : HTAB 12:
       PRINT "COPY PROTECT DISK
      ": VTAB 23: HTAB 3: NORMA
      L : PRINT "[ESC] TO GO BA
      CK TO THE MAIN MENU"
2E 300 VTAB 10: HTAB 12: PRINT "
      COMBINATION LOCKS": PRINT
       : PRINT "      #1      #
      2      #3      #4"
22 310 PRINT "      ( )      ( )
          ( )      ( )"
60 320 FOR NL = 1 TO 4: VTAB 13:
       HTAB B * NL - 1: GET A$
D1 330 IF A$ = CHR$ (27) THEN RU
      N
AC 340 IF VAL (A$) > 9 THEN NL =
       NL - 1: NEXT NL
67 350 IF VAL (A$) = 0 AND A$ <
      > "0" THEN NL = NL - 1: N
      EXT NL
24 360 CL(NL) = VAL (A$): PRINT
      CL(NL): NEXT NL
D0 370 POKE 34,19
9D 380 VTAB 20: PRINT "    INSERT
       SOURCE DISK INTO DRIVE 1
      ": HTAB 4: PRINT "PRESS [
      RETURN] TO BEGIN PROCESS"
56 390 VTAB 24: HTAB 20: GET A$
CC 400 IF A$ = CHR$ (27) THEN RU
      N
87 410 IF A$ < > CHR$ (13) THEN
      GOTO 390
D5 420 POKE 47520,134: POKE 4752
      1,43: POKE 47522,133: POK
      E 47187,213: POKE 47335,2
      13: POKE 48250,213: POKE
      47445,213:SS = 9:SE = 9:O
      P = 1:BU = 8192:TR = 1: G
      OSUB 710
95 430 A$ = "": FOR A = 117 TO 1
      54:A$ = A$ + CHR$ ( PEEK
      (B192 + A)): NEXT A
55 440 POKE 47520,76: POKE 47521
      ,141: POKE 47522,183
98 450 V1 = CL(1) * 10 + CL(2):V
      2 = CL(3) * 10 + CL(4)
04 460 IF V1 = 31 THEN V1 = 121
8E 470 IF V1 = 63 OR V1 = 64 THE
      N V1 = V1 + 50
79 480 IF V2 = 31 THEN V2 = 121
35 490 IF V2 = 63 OR V2 = 64 THE
      N V2 = V2 + 50
7A 500 POKE 48357,V1 + 129: POKE
       48359,V2 + 129
22 510 HOME : VTAB 24: HTAB 2: P
      RINT "INSERT DESTINATION
      DISK INTO DRIVE 1": HTAB
      12: PRINT "AND PRESS A KE
      Y": HTAB 20: GET B$
69 520 HOME : HTAB 5: FLASH : PR
      INT "INITIALIZING";: NORM
      AL : PRINT " DESTINATION
      DISK"
4E 530 PRINT : PRINT CHR$ (4);"I
      NIT ";A$;",D1"
6C 540 HOME : HTAB 4: PRINT "INS
      ERT ORIGINAL DISK INTO DR
      IVE 1": HTAB 12: PRINT "A
      ND PRESS A KEY": HTAB 20:
       GET B$
40 550 TC = 0:BF = B192: FOR TK
      = 3 TO 34:TC = TC + 1: VT
      AB 6: HTAB 16: PRINT "TRA
      CK: ";TK: HOME : HTAB 7:
      PRINT "READING FROM ORIGI
      NAL DISK":BF = BF + 4096
40 560 POKE 796,1: POKE 7BB,TK:
      POKE 7B9,15: POKE 792,0:
      POKE 793, INT (BF / 256)
65 570 POKE 47520,134: POKE 4752
      1,43: POKE 47522,133: POK
      E 47187,213: POKE 47335,2
      13: POKE 48250,213: POKE
      47445,213
A1 580 CALL 7000
9D 590 IF TK = 7 OR TK = 12 OR T
      K = 17 OR TK = 22 OR TK =
       27 OR TK = 32 THEN GOTO
      610
34 600 NEXT TK
7D 610 HOME : HTAB 2: PRINT "INS
      ERT DESTINATION DISK INTO
       DRIVE 1": HTAB 12: PRINT
       "AND PRESS A KEY": HTAB
      20: GET A$
00 620 IF TK = 35 THEN TK = 34
CF 630 BF = 8192: FOR TA = TK -
      TC + 1 TO TK: VTAB 6: HTA
      B 16: PRINT "TRACK: ";TA;
      " ": HOME : HTAB 7: PRINT
       "WRITING TO DESTINATION
      DISK":BF = BF + 4096
C2 640 POKE 796,2: POKE 788,TA:
      POKE 789,15: POKE 792,0:
      POKE 793, INT (BF / 256)
78 650 POKE 47520,76: POKE 47521
      ,141: POKE 47522,183: POK
      E 48357,V1 + 129: POKE 48
      359,V2 + 129
9E 660 CALL 7000
38 670 NEXT TA
01 680 IF TA = 35 THEN HOME : PR
      INT "---DONE---": GET B$:
       RUN
2A 690 HOME : HTAB 3: PRINT "INS
      ERT ORIGINAL DISK INTO DR
      IVE 1": HTAB 12: PRINT "A
      ND PRESS A KEY": HTAB 20:
       GET B$:BF = 8192:TC = 0:
       NEXT TK
0F 700 END
5E 710 REM ***DISK ACCESS***
07 720 FOR SA = SS TO SE
F9 730 POKE 7BB,TR: POKE 789,SA:
       POKE 796,OP
C9 740 HB = INT (BU / 256):LB =
      BU - (HB * 256)
78 750 POKE 792,LB: POKE 793,HB
29 760 CALL 768:BU = BU + 256: N
      EXT SA
25 770 RETURN
```

©

Directory Plus For Commodore

Thomas C. Carlson

This utility program prints a comprehensive disk directory on the screen or a printer, giving you extra information about the files on your disks. The program requires a 1541 or 1571 disk drive and runs on the Commodore 64, 128, Plus/4, 16, and VIC-20 (with at least 8K expansion). A printer is optional.

Virtually every Commodore disk drive owner knows how to get a listing of a disk directory. The statement LOAD"$0",8 loads the directory into memory, and LIST displays it on the screen. To print the directory on a printer, type OPEN 4,4 before you load the directory, and PRINT#4:CLOSE 4 after the listing is complete. The normal directory listing—which includes the filename of each file, its file type, and number of blocks—is fine for everyday use, but inadequate for more advanced purposes. In many programming situations it is necessary to know the load address of a file or the actual track and sector where it begins. When many files are involved, discovering such information can be a tedious process.

"Directory Plus" solves this by automatically printing an expanded disk directory on the screen or printer. In addition to the usual information, the expanded directory includes the disk track and sector where the file begins, and the load address of the file (the address where the file usually loads into the computer's memory). The accompanying figure illustrates a Directory Plus printout for a typical COMPUTE! DISK.

Directory Plus Printout

C! MAY-JULY 1988		MJ	2A	FR SEC:	206
FILENAME:	TYP	TR	SC	BLK	START
MENU	PRG	17	0	1	812
128 BOOK8	PRG	17	1	20	2049
128.BOOT	PRG	0	3	6	2049
64 BOOK8	PRG	18	0	32	2049
64 CONTENTS	SEQ	14	0	5	
ALL ABOUT THE 64	PRG	21	2	83	2049
AUTOBOOTER	PRG	14	4	12	2049
COMPUTE!	PRG	16	0	2	2049
COMPUTECOLOR	PRG	16	1	4	55296
COMPUTESCREEN	PRG	16	3	4	1024
CUBE1	PRG	6	0	1	2049
CUBE	PRG	29	9	1	2049
FLEET LIST.BOOT	PRG	30	10	1	2049
FLEET LIST	PRG	13	0	4	49152
GAZETTE	PRG	16	5	13	2049
HEX WAR/128	PRG	0	0	39	10369
HEX WAR/64.BOOT	PRG	29	4	1	16385
HEX WAR/64	PRG	26	0	37	10385
HICKORY DICKORY	PRG	13	1	28	2049
LG	PRG	12	1	7	49152
LOOK GLASS.BOOT	PRG	30	0	1	2049
LOOKINGGLASSDEMO	PRG	12	0	12	2049
MANDELBROT 1	PRG	28	9	12	2049
MANDELBROT 2	PRG	7	9	2	2049
MANDELBROT 3	PRG	29	0	3	2049
MANDELBROT 4	PRG	29	2	2	2049
MANDELBROT.BOOT	PRG	30	11	4	2049
MIAMI ICE/128	PRG	10	0	47	7169
MIAMI ICE/64	PRG	11	2	12	2049
ML DIVISION.BOOT	PRG	30	16	4	2049
ML DIVISION	PRG	29	19	2	2049
MLX	PRG	28	7	17	2049
MN	PRG	19	0	8	2049
NT	PRG	14	1	3	2049
PROGFREADER	PRG	15	1	8	7169
RAM REPORT	PRG	6	1	3	2049
SCR HANDLER DEMO	PRG	28	1	3	2049
SCR HANDLER.BOOT	PRG	30	19	1	2049
SCREEN HANDLER	PRG	28	3	4	49152
SEQ FILE CONVERT	PRG	6	0	3	2049
SQUARE1	PRG	8	7	1	2049
SQUARE	PRG	23	9	1	2049
UPSTART	PRG	6	12	4	2049

Directory Plus works without modification on the Commodore 64, 128 (40- or 80-column screens), Plus/4, 16, and VIC-20 (with at least 8K exapnsion). Since the VIC-20 screen has only 22 columns, its directory display is less neatly formatted than the others; however, the printer output is exactly the same for all versions.

Program Setup

After you have entered and saved a copy of Directory Plus, run the program. It begins by asking whether you want to display the directory on the screen or a printer. Press S for screen output or P for printer output.

If you're using a printer, be sure it is connected properly and turned on before proceeding any further. Directory Plus is designed to work with the following Commodore printers: MPS-801, MPS-802, MPS-803, 1525, and 1526. As listed below, the program is set up to work with the MPS-802 and 1526 printers. If you have an MPS-801, MPS-803 or 1525 printer, remove the keyword REM from the beginning of line 20 (but leave the rest of the line intact).

The program also works as is with non-Commodore printers, *but only if your printer/interface combination can emulate Commodore graphics mode exactly.* In this case, you should *probably* remove the REM in line 20 to activate the Commodore graphics mode; however, some interfaces for non-Commodore printers may require that you send additional codes to the interface to put it in Commodore graphics mode. It may also be necessary to add a secondary address to the OPEN statement in line 790. Consult the manuals for your printer and interface if you are in any doubt about the capabilities of your system.

The program can easily be modified to work with printers that

do not support Commodore graphics as well. Simply replace the graphics characters in lines 800–910 with spaces, or use dashes, asterisks, or any other characters you wish.

If you select the printer option when displaying the directory, a second prompt will appear asking you to select the printing width. Press S for a single-width (normal) printout, or D for a double-width printout. Many printer interfaces that support Commodore graphics do not support the graphics characters in double-width mode, so you may not be able to use the D option if you have a non-Commodore printer.

Load Addresses

At this stage the program prompts you to insert the disk whose directory you wish to view. Press any key when the disk is in place. After a pause while the computer reads the disk directory, the program asks whether you want to see the load addresses of any files. To display the directory without any address information, press the 3 key. If you want to see the load address for every file on the disk, press the 1 key. To view load addresses for only selected files, press 2. When this option is selected, the program displays each filename in turn, allowing you to choose whether you want to see its load address; press Y to display the load address of the current file, or N to skip to the next file. Note that some files (data files, for instance) don't contain a meaningful load address. In such cases, no address is displayed.

If you choose to display load addresses, the disk drive spins for a few moments while it retrieves this extra information for each file. You should not continue past this stage until the drive is finished working (when using the 1541 drive, wait until the motor stops spinning; on the 1571, wait until the drive's busy light goes off).

After every prompt has been answered and the drive is at rest, the directory display begins. To slow the scrolling of screen output, hold down the CTRL key on the VIC or 64, the Commodore key on the Plus/4 or 16, or CTRL-S on the 128.

After the directory has been printed on the screen or printer, Directory Plus gives you the option of viewing the same directory again, or of changing disks and printing a directory for the new disk.

Directory Plus does not display information about deleted (DEL) type files. DEL files are rarely of interest; however, if you wish to view them, delete line 870 from the program. Another possible modification involves the drive number. Although the 1541 and 1571 drives are always addressed as drive 0, some Commodore-compatible dual drives include drive 1 as well as drive 0. To access drive 1 in a dual drive system, change the 0 to a 1 in lines 170, 180, and 640.

In general, Directory Plus works by opening the directory as a sequential file and bringing in the contents one character at a time with the GET statement. The manual that came with your disk drive contains additional information about the structure of the directory. For those interested in writing similar programs, here is a brief outline of the major segments in Directory Plus:

Lines	Function
10–160	Initialization
170–230	Read header information
240–480	Read file information
490–700	Read load addresses
710–930	Print directory listing
940–1010	Repeat options
1020–1050	Check error channel

Directory Plus

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!

```
OP 10 ND$=""
AK 20 REM NO$=CHR$(8):REM REMO
      VE REM FOR 1525 OR MPS-8
      01 PRINTERS
DM 30 OIM FT$(5):FORI=0TO5:REA
      D A$:FT$(I)=A$:NEXT
X8 40 DATA OEL,SEQ,PRG,USR,REL
      ,OEL
BA 50 OIM F$(144,5)
XE 60 FE=664
DC 70 PRINTCHR$(147);CHR$(30);
      CHR$(17);"OUTPUT TO SCRE
      EN OR PRINTER (S/P) ?"
8H 80 GETA$:IFA$=""GOTO80
KJ 90 DV=3:IFA$="P"THENOV=4
QQ 100 IFA$="S"THENGOTO140
KF 110 PRINTCHR$(17)"SINGLE OR
       DOUBLE WIDTH (S/O) ?"
KD 120 GETA$:IFA$=""GOTO120
AG 130 BG$=CHR$(15):IFA$="O"TH
      ENBG$=CHR$(14)
AS 140 PRINTCHR$(17)"INSERT DI
      SK ANO PRESS ANY KEY"
RE 150 GET A$:IF A$="" THEN150
DX 160 PRINTCHR$(145)"PLEASE W
      AIT...{15 SPACES}"
EC 170 OPEN15,8,15:PRINT#15,"I
      0":GOSUB1020
RS 180 OPEN8,8,8,"$0,S,R"
SX 190 GOSUB1020
CR 200 FORI=1TO142:GET#8,A$:NE
      XT
AB 210 FORI=143TO160:GET#8,A$:
      N$=N$+A$:NEXT
FG 220 FORI=161TO162:GET#8,A$:
      IO$=ID$+A$:NEXT
CP 230 GET#8,A$:FORI=164TO165:
      GET#8,A$:OS$=OS$+A$:NEX
      T
EH 240 FORI=166TO254:GET#8,A$:
      NEXT
PQ 250 CT=8
SF 260 NM=NM+1
PX 270 IFCT=8THENCT=1:GOTO300
RH 280 CT=CT+1:GET#8,A$,A$:FL=
      ST
GP 290 IFFL<>0GOTO480
EG 300 GET#8,A$:IFA$=""THENA$=
      CHR$(133)
XA 310 FL=ST:IFFL<>0GOTO480
MX 320 TY$=FT$((ASC(A$)AND191)
      -128)
GO 330 GET#8,A$:IFA$=""THENA$=
      CHR$(0)
ME 340 TR$=RIGHT$("{2 SPACES}"
      +STR$(ASC(A$)),2)
S8 350 GET#8,A$:IFA$=""THENA$=
      CHR$(0)
88 360 SC$=RIGHT$("{2 SPACES}"
      +STR$(ASC(A$)),2)
DM 370 FL$="":FORI=3TO18:GET#8
      ,A$:FL$=FL$+A$:NEXT
SF 380 FORI=19TO27:GET#8,A$:NE
      XT
SC 390 GET#8,L8$,H8$
AE 400 BL=ASC(L8$+CHR$(0))+256
      *ASC(HB$+CHR$(0))
MS 410 IFTY$<>"DEL"THENFE=FE-8
      L
FC 420 BL$=RIGHT$("{5 SPACES}"
      +STR$(BL),3)
HC 430 IFTR$=" 0"GOTO480
DB 440 F$(NM,0)=FL$:F$(NM,1)=T
      Y$:F$(NM,2)=TR$:F$(NM,3
      )=SC$:F$(NM,4)=BL$
OK 450 F$(NM,5)="{5 SPACES}"
PH 460 IFTY$="PRG"THENF$(NM,5)
      ="-----"
KC 470 GOTO260
GR 480 CLOSE8
XA 490 GOSUB1020
HR 500 IFF$(NM,0)=""THENNM=NM-
      1:GOTO500
SK 510 FE$=RIGHT$("{4 SPACES}"
      +STR$(FE),3)
XQ 520 PRINTCHR$(145);"START A
      DORESS :":PRINTCHR$(17)
      ;" 1) ALL"
MJ 530 PRINT" 2) SOME":PRINT"
      {SPACE}3) NONE"
EK 540 GETA$:IFVAL(A$)=0THENGO
      TO540
DS 550 IFVAL(A$)>2GOTO770
MD 560 PRINT"{CLR}":IFA$="1"TH
      ENPRINT"JUST A MOMENT..
      ."
8Q 570 FORI=1TONM
MQ 580 IFF$(I,1)<>"PRG"GOTO700
XK 590 IFA$="1"GOTO630
BS 600 PRINTF$(I,0);" (Y/N)"
PX 610 GETA$:IFA$=""GOTO610
RB 620 IFA$<>"Y"THENPRINT"{UP}
      ";:GOTO700
QJ 630 SA$=F$(I,0)
EK 640 OPEN 8,8,8,"0:"+SA$+",P
      ,R"
```

```
8M 650 GOSU81020
QB 660 GET#8,L8$,H8$
SH 670 SA=ASC(LB$+CHR$(0))+256
       *ASC(HB$+CHR$(0))
PM 680 CLOSE8
AK 690 F$(I,5)=RIGHT$("
       {6 SPACES}"+STR$(SA),5)
MR 700 NEXT
GK 710 PRINT"{3 DOWN}WAIT UNTI
       L THE DRIVE LIGHT GOES
       {SPACE}OFF"
EB 720 PRINT CHR$(28);SPC(21);
       :FORQQ=1TO14:PRINT CHR$
       (163);:NEXT:PRINT CHR$(
       30);
ME 730 PRINT:PRINT"THEN ";
EC 740 IFDV=4THENPRINT"SET PRI
       NTER & ";
EH 750 PRINT"PRESS ANY KEY"
JE 760 GET A$:IF A$="" THEN760
SA 770 IFDV<>4THENPRINTCHR$(14
       7)
AP 780 IFDV=4THENIFND$=""THENO
       PEN6,4,6:PRINT#6,CHR$(2
       1):CLOSE6
FH 790 OPEN4,DV
BJ 800 PRINT#4,BG$;"{A}*******
       **********{R}**{R}**
       {R}**********{S}";ND$
AA 810 PRINT#4,BG$;"-";N$;"-";
       ID$;"-";OS$;"-FR SEC: "
       ;FE$;"-";ND$
GR 820 PRINT#4,BG$;"{Q}*******
       *********{R}*{E}*{R}{E}
       *{R}{E}*{R}***{R}*****
       {W}";ND$
JQ 830 PRINT#4,8G$;"-FILENAME:
       {7 SPACES}-TYP-TR-SC-BL
       K-START-";ND$
GR 840 PRINT#4,8G$;"{Q}*******
       *********+***+**+**+***
       +*****{W}";ND$
RS 850 FORI=1TONM
CF 860 FL$=F$(I,0):TY$=F$(I,1)
       :TR$=F$(I,2):SC$=F$(I,3
       ):8L$=F$(I,4)
JB 870 IFTY$="DEL"GOTO900
XK 880 PRINT#4,BG$;"-";FL$;"-"
       ;TY$;"-";TR$;"-";SC$;"-
       ";BL$;"-";F$(I,5);"-";N
       D$
MQ 890 PRINT#4,8G$;"-
       {16 SPACES}-{3 SPACES}-
       {2 SPACES}-{2 SPACES}-
       {3 SPACES}-{5 SPACES}-"
       ;ND$
BK 900 NEXT
GR 910 PRINT#4,8G$;"{Z}*******
       *********{E}***{E}**{E}
       **{E}***{E}*****{X}";ND
       $
AS 920 PRINT#4:CLOSE4
MH 930 CLOSE 15
HJ 940 IFDV=4THENPRINT"{CLR}PR
       INT AGAIN (Y/N) ?"
GM 950 IFDV=3THENPRINT"VIEW AG
       AIN (Y/N) ?"
8C 960 GETA$:IFA$="Y"THEN770
RM 970 IFA$<>"N"THEN960
JR 980 PRINTCHR$(145)"NEW DIRE
       CTORY (Y/N) ?{6 SPACES}
       "
8D 990 GETA$:IFA$="Y"THENRUN
X8 1000 IFA$="N"THEN PRINT"
       {CLR}":END
GD 1010 GOTO990
BQ 1020 INPUT#15,EZ,EZ$,TR,SE:
       IF EZ=0 THEN RETURN
KD 1030 T$=CHR$(157)+CHR$(32)
CK 1040 PRINT CHR$(18) EZ;T$;E
       Z$;TR;T$;SE
HE 1050 CLOSE 8:CLOSE 15      ©
```

The Logical Alternative:

True-False Logic in Atari BASIC

Ronald R. Lambert

As this article demonstrates, there's a compact and efficient alternative to conventional IF-THEN statements: logical comparisons. The techniques described here work with Atari BASIC on the 400/800, XL, and XE computers—and, with slight adjustments, with all versions of BASIC.

Anyone who has read a BASIC reference manual knows about logical operators such as >, <, =, AND, and OR. These are most commonly used in IF-THEN statements:

```
IF X>0 THEN PRINT X
```

(X will be printed only if it is greater than zero.)

But there is another way to use logical statements, one that can streamline and shorten programs considerably—especially in Atari BASIC, which allows calculated GOTOs, GOSUBs, and RESTOREs.

BASIC tests logical statements to see if they are true or false. In keeping with the principles of Boolean algebra, the value 1 is applied to a statement if it is true, and a 0 is applied if the statement is false. (Some BASICs, such as those found on Commodore computers, the IBM PC and PCjr, and Texas Instruments TI-99/4A, apply a −1 if the statement is true.) When the value is true, the statement following the THEN clause in an IF-THEN statement is executed. When the value is false, the program skips to the next line. (Note that the latest BASICs usually let you add an optional ELSE clause to an IF-THEN statement. Execution would then continue with the statement following ELSE.)

The same true-false evaluation also happens with any logical BASIC statement, such as X=10. Taken by itself (this may require enclosing the statement in parentheses), a statement like X=10 can be used as a variable—a variable that can equal 1 or 0, depending on whether the equation is true or not. Let's see how we can take advantage of this to shorten a program line.

Logic Versus IF-THEN

Instead of this:

```
100 IF X=10 THEN Y=Y+1
```

Try this:

```
100 Y=Y+(X=10)
```

If you're using a BASIC that assigns a −1 to true statements, change the sign of the statement:

Y=Y−(X=10). Subtracting −1 is the same as adding 1.

Both of the above statements mean the same thing and will accomplish the same function: Y is incremented only if X=10. In the second example, IF-THEN is replaced by a logical evaluation. If X *does not* equal 10, then the statement (X=10) has an assigned value of 0, and 0 is added to Y—leaving the value of Y unchanged. Only when X *does* equal 10 will the statement have a value of 1, causing the value of Y to be incremented.

Not only is the second example shorter, but notice the way it is constructed—the program will not skip to the next line if X does not equal 10, but instead can continue on to read further statements in the same program line. In fact, several IF-THEN statements in effect can be combined into one line, as the following two examples demonstrate.

Instead of this:

```
10 X=X+1:IF X=255 THEN Y=Y+1:
   X=0
20 IF Y=255 THEN Z=Z+1:Y=0
30 IF Z=255 THEN PRINT "DONE"
   :END
40 GOTO 10
```

Try this:

```
10 X=X+1:Y=Y+(X=255):X=X-255*
   (X=255):Z=Z+(Y=255):Y=Y-255*
   (Y=255):IF Z<255 THEN 10
20 PRINT "DONE":END
```

(Remember, if you're using a BASIC that assigns −1 to true statements, reverse the signs in the latter example, except for the statement X=X+1.)

Again, both of the above examples do the same things. They increment Y by 1 every time the value of X reaches 255 (and also reset X to 0), increment the value of Z every time the value of Y reaches 255 (and reset Y to 0), and then when the value of Z reaches 255, print the message DONE.

In the second example, where logic is used, the statement (X=255) is multiplied by 255 and subtracted from X. As long as X *does not* equal 255, the value of the statement will be zero. Since 255 times 0 is 0, then 0 is what is subtracted from X, leaving the value of X unchanged. But when X equals 255 and the equation is true, then we have 255 times 1 (or −1, depending on your computer), which is 255. If X equals 255, then subtracting this value from X changes the value of X to 0. (If you're using a BASIC that assigns −1 to true statements and have changed the signs in the above statements as noted, then −255 will be added to X when X equals 255. Adding a negative number is the same as subtracting.)

The same is true for the statement Y=Y−255*(Y=255). In effect, four conditional statements have been combined into one line.

Logical Branching

As mentioned earlier, Atari BASIC allows calculated GOTOs, GOSUBs, and RESTOREs. When logical statements are used in these calculations, it is possible to branch to any line in the program depending upon which logical statement is true. This can save substantial amounts of memory. Consider the following program:

```
10 OPEN #2,4,0,"K:"
20 ? "Type S, L, or P.":GET #2,N:IF
   N=83 THEN 60
30 IF N=76 THEN 70
40 IF N=80 THEN 80
50 GOTO 20
60 ? "This could be a save to tape or disk
   routine.":GOTO 20
70 ? "This could be a load from tape or
   disk routine.":GOTO 20
80 ? "This could be an output to printer
   routine.":GOTO 20
```

If we use logic, the program can be substantially shortened. Delete lines 30, 40, and 50, and replace line 20 with this:

```
20 ? "Type S, L, or P.":GET #2,N:GOTO
   20+40*(N=83)+50*(N=76)+60*
   (N=80)
```

The program works exactly the same as before.

Timing Tradeoffs

Substituting logical statements for IF-THEN statements usually slows down an Atari BASIC program, though normally the difference is too slight to matter, especially when the line is executed only once or just a few times. But the difference is measurable when the statements are enclosed in loops.

To demonstrate, following is a short program that counts words in a long text string. (Actually it counts spaces, an easy way to get a fairly accurate word count.) It generates a long string of text, then uses two of the Atari's internal clock registers to time the two methods for counting. First the words are counted using a conventional IF-THEN construction, and then they're counted using logic.

```
10 DIM TEXT$(2362)
20 TEXT$="Welcome to the
   Overlook Hotel. All wo
   rk and no play makes J
   ack a dull boy."
30 TEXT$(2318)=TEXT$(31):
   TEXT$(75)=TEXT$(31):?
   TEXT$
40 ? :? "Counting...":WOR
   OCOUNT=0:POKE 19,0:POK
   E 20,0
50 FOR X=1 TO LEN(TEXT$)
60 IF ASC(TEXT$(X))=32 TH
   EN WOROCOUNT=WORDCOUNT
   +1
70 NEXT X:? WORDCOUNT;" w
   ords counted using IF-
   THEN in ";PEEK(19)*256
   +PEEK(20);" jiffies (i
   nternal timer)."
80 ? :? "Counting...":WOR
   OCOUNT=0:POKE 19,0:POK
   E 20,0
90 FOR X=1 TO LEN(TEXT$)
100 WOROCOUNT=WORDCOUNT+(
    ASC(TEXT$(X))=32)
110 NEXT X:? WOROCOUNT;"
    words counted using l
    ogic in ";PEEK(19)*25
    6+PEEK(20);" jiffies
    (internal timer)."
```

When you type in and run this program, it displays for you the word counts and the time required for each count measured in *jiffies*, which are equal to 1/60 second. In this case, the IF-THEN routine (line 60) runs a little faster than the logical statement equivalent (line 100).

Now that you know how logical statements work, you may take a shine to the kind of programming techniques they make available. They certainly provide a logical alternative. ©

COMPUTE!
TOLL FREE
Subscription
Order Line
1-800-247-5470
In IA 1-800-532-1272

Commodore *SpeedScript* To BASIC

Frank Colosimo
Mike Kozakiewicz

This utility program provides a convenient way to convert text in a SpeedScript *word processing file into BASIC PRINT or DATA statements. The result is a BASIC program which you can load and run as a stand-alone program or add to existing programs of your own. The utility program requires a Commodore 64 or 128 (in 64 mode), a copy of Commodore 64* SpeedScript, *and a disk drive.* SpeedScript *was published in the March 1985 issue of* COMPUTE! *and also is available in* SpeedScript: The Word Processor for the Commodore 64 and VIC-20 *from COMPUTE! Books.*

One of the first commands a BASIC programmer learns to use is PRINT, yet no matter how advanced you become, formatting a text display with PRINT can involve a lot of trial and error. If you PRINT past the right edge of the screen, words may break in the middle rather than wrapping completely around to the next line. And changing just one PRINT statement can affect the appearance of an entire screen.

"*SpeedScript* to BASIC" provides an answer for anyone who wants an easy way to format text neatly on the screen. It takes a text file created with Commodore 64 *SpeedScript* and converts it into PRINT or DATA statements ready to be merged with your own program. Some uses for *SpeedScript* to BASIC include creating instruction screens for BASIC programs, preparing self-contained educational or advertising programs, or converting word processing files into BASIC programs that can be read without the use of a word processor.

If you're a nonprogrammer, you may find it particularly useful for turning word processing files into BASIC programs. The programs it automatically generates are completely self-contained and display the text onscreen without use of the word processor itself.

Format Without Frustration

Type in the program listing below, then save a copy to disk. Before you can use the program, you must create a text file for it to process. Load and run *SpeedScript*, then type in as much text as you wish. When that's done, save the *SpeedScript* document as usual, then exit the word processor and load and run this program.

The program begins by asking you whether you want its output in the form of DATA statements or PRINT statements. The answer depends on your goal. The PRINT option is most useful if you intend to add the resulting display routine to an existing program of your own. If you want a stand-alone program, choose the DATA option; this creates an independent program that will display formatted text, one screen at a time, as you press a key.

After choosing the output type, you are asked for the name of the input file. Enter the filename of your previously prepared *SpeedScript* file, then press RETURN. If you're not sure of the exact filename, you can enter a dollar sign ($) to view the disk directory. If you ask for a file that does not exist, the program lets you try again. Enter Q at this prompt if you want to end the program.

The program now reads your word processing file and constructs a series of new BASIC statements in a large buffer area within memory. The file conversion routine is written in machine language for maximum speed. To keep you updated, the program increments the counter display each time it processes another 256 characters of text.

Once the work is done, the program asks you to insert an output disk in the drive. You then enter a name for the output program file to be created. If the file already exists on your output disk, you are asked if you want to erase the existing file. If you choose not to erase, you are asked to enter a new filename. You may also end the program by entering Q at this prompt. The output file is then saved to disk, and you are given the opportunity to save a copy to another disk.

Accurate Reproduction

The result is a set of BASIC program lines which accurately recreate the original text display. Just as in *SpeedScript*, the program wraps words instead of splitting them at the right edge of the screen. It also ignores *SpeedScript* formatting codes, which are relevant only when printing a document on paper. All other characters are faithfully reproduced, except for

quotation marks. Since the PRINT command itself requires the use of double quotes, the program substitutes a single quote wherever a double quote appears in the original text.

Once you learn how easy it is to use, you'll probably find more and more uses for this program. To simplify the job of adding the new program lines to existing programs, the line numbers start at line 10000 and use increments of 10. A 21-block word processing file takes about 38 seconds to be processed into a 26-block BASIC program (not counting the time you spend answering the prompts).

If you're interested in examining the machine language routine that makes this utility work, you can find it at locations 49152 and above.

SpeedScript To BASIC

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!.

```
DF 10  IFPEEK(49152)=76THENGOTO
       40
KD 20  PRINT"{CLR}{2 SPACES}PLE
       ASE WAIT, STORING DATA..
       ."
HX 30  FOR M = 49152 TO 50153:
       {SPACE}READ A:POKEM,A:NE
       XT
JM 40  POKE53281,15:POKE53280,0
GK 50  POKE56,PEEK(46)+2:CLR
RG 60  B$="{40 SPACES}"
AS 70  PRINT"{CLR}{BLK}{RVS}{N}
       {3 SPACES}*****
       {2 SPACES}SPEEDSCRIPT TO
        8ASIC{2 SPACES}*****
       {3 SPACES}"
FD 80  LD%=2:GOSU8500:INPUT"DAT
       A OR PRINT STATEMENTS (D
       /P){3 SPACES}D{3 LEFT}";
       TYPE$
QH 90  TYPE$=LEFT$(TYPE$,1):IF(
       TYPE$<>"D") AND (TYPE$<>
       "P") THEN GOTO80
JD 100 ADDR=49152:IF TYPE$="P"
        THEN ADDR=49155
KA 110 CLOSE15:OPEN15,8,15,"I0
       "
SM 120 LD%=2:GOSU8500
KR 130 PRINT"{DOWN}INPUT FILEN
       AME, $ (DIR), OR Q (QUI
       T)"
AP 140 INPUT"{3 SPACES}Q
       {3 LEFT}";IN$:PRINT"
       {CLR}"
CJ 150 IF IN$<>"$"ANDIN$<>"Q"
       {SPACE}THEN PRINT"{CLR}
       {DOWN}{12 RIGHT}READING
        FILE..."
XQ 160 IFIN$=""THEN120
8G 170 IF IN$="$"THENSYS49994:
       GOTO130
QF 180 IF IN$="Q"THEN GOTO480
MX 190 LD%=2:GOSUB500
HK 200 CLOSE1:OPEN1,8,3,IN$:IN
       PUT#15,EN,EM$:F$=IN$:IF
       EN=0THEN220
JS 210 GOSUB550:GOTO120
CE 220 SYS(ADDR):CLOSE1:SYS654
       84
QM 230 PRINT"{CLR}":LD%=11:GOS
       U8500:PRINT"{10 SPACES}
       {RVS}INSERT OUTPUT DISK
       {OFF}"
BJ 240 GOSU8520:ADDR=49158
CD 250 LD%=2:GOSUB500
JH 260 PRINT"{DOWN}OUTPUT FILE
        NAME, $ (DIR), OR Q (Q
       UIT)"
PC 270 INPUT"{2 SPACES}Q
       {3 LEFT}";OUT$:PRINT"
       {CLR}"
BC 280 IF OUT$="Q"THEN GOTO480
KG 290 IFOUT$="$"THENSYS49994:
       GOTO260
PG 300 LD%=4:GOSU8500
FJ 310 IF IN$<>"$"ANDIN$<>"Q"
       {SPACE}THEN PRINT"{CLR}
       {DOWN}{13 RIGHT}WRITING
        FILE..."
QE 320 CLOSE9:OPEN9,8,4,"0:"+O
       UT$:INPUT#15,EN,EM$:F$=
       OUT$:CLOSE9
8X 330 IFEN=0THEN360
RA 340 IFEN<>62THEN GOSUB550:G
       OTO230
RJ 350 GOTO400
BB 360 LD%=6:GOSU8500:PRINTOUT
       $;" EXISTS... REPLACE?
       {SPACE}{RVS}Y{OFF}/
       {RVS}N{OFF};"
RG 370 GETA$:IFA$<>"Y"ANDA$<>"
       N"THEN370
PA 380 IFA$="N"THEN230
SD 390 PRINT#15,"S0:"+OUT$
QS 400 LD%=15:GOSU8500
RM 410 T1=8+LEN(OUT$):B1$=LEFT
       $(8$,(20-(T1/2))):PRINT
       "{CLR}"81$;"{RVS}SAVING
       : ";OUT$;"{OFF}"
GS 420 GOTO440
EJ 430 PRINT"{CLR}"
RE 440 SYS(ADDR),OUT$:SYS65484
       :PRINT"DONE
DJ 450 INPUT"MAKE ANOTHER COPY
       {3 SPACES}N{3 LEFT}";AN
       $
GM 460 AN$=LEFT$(AN$,1):IF(AN$
       <>"Y")AND(AN$<>"N")THEN
       450
KJ 470 IF AN$="Y"THEN GOTO230
JQ 480 POKE56,160
FE 490 FORI=1TO15:CLOSEI:NEXT:
       CLR:PRINT"{CLR}":END
MB 500 SYS49161,LD%
GP 510 PRINT"{HOME}":FORJ=1TOL
       D%-1:PRINT:NEXT:RETURN
RD 520 PRINT"{DOWN}{8 SPACES}
       {RVS}PRESS A KEY TO CON
       TINUE"
FC 530 GETAN$:IFAN$=""THEN530
QJ 540 RETURN
CH 550 PRINT"{DOWN}DISK ERROR
       {SPACE}FOR ";F$:PRINTEM
       $:GOSUB520:RETURN
CD 560 DATA 76,12,192,76,34,19
       2,76,239,194,76,246,194
FS 570 DATA 169,32,141,197,193
       ,169,218,141,198,193,16
       9,193
CQ 580 DATA 141,199,193,169,13
       1,141,242,192,208,15,16
       2,2
SC 590 DATA 169,234,157,197,19
       3,202,16,250,169,153,14
       1,242
QE 600 DATA 192,32,168,193,238
       ,68,192,208,13,32,164,1
       94
8D 610 DATA 162,1,32,198,255,7
       6,70,192,255,0,32,152
FS 620 DATA 193,133,2,32,183,2
       55,41,64,240,3,76,0
JM 630 DATA 193,32,91,192,76,5
       2,192,165,2,48,31,32
KM 640 DATA 68,193,201,13,208,
       4,32,98,193,96,201,32
MG 650 DATA 208,4,32,123,193,9
       6,32,127,192,165,20,201
C8 660 DATA 39,208,3,32,98,193
       ,96,164,20,153,135,192
KA 670 DATA 230,20,96,0,0,0,0,
       0,0,0,0,0
GQ 680 DATA 0,0,0,0,0,0,0,0,0,
       0,0,0
CX 690 DATA 0,0,0,0,0,0,0,0,0,
       0,0,0
RA 700 DATA 0,0,0,0,0,0,32,156
       ,193,230,21,96
MS 710 DATA 166,20,240,17,162,
       0,189,135,192,32,174,19
       2
AF 720 DATA 232,228,20,208,245
       ,162,0,134,20,96,32,212
GG 730 DATA 193,32,216,192,32,
       180,192,96,169,0,133,20
PC 740 DATA 169,10,32,156,193,
       32,156,193,24,101,158,1
       33
ES 750 DATA 158,144,2,230,159,
       32,156,193,165,159,32,1
       56
MG 760 DATA 193,169,131,32,156
       ,193,169,34,32,156,193,
       169
CD 770 DATA 0,133,21,96,32,91,
       192,32,98,193,169,29
RB 780 DATA 32,156,193,169,157
       ,32,156,193,32,212,193,
       32
HM 790 DATA 212,193,32,212,193
       ,96,169,2,162,8,160,1
RR 800 DATA 32,186,255,32,253,
       174,32,158,173,160,0,17
       7
BS 810 DATA 71,72,200,177,71,1
       70,200,177,71,168,104,3
       2
XS 820 DATA 189,255,169,55,164
       ,196,166,195,32,216,255
       ,96
GJ 830 DATA 41,64,10,5,2,41,19
       1,133,2,41,32,73
CJ 840 DATA 32,10,5,2,201,95,2
       08,3,169,13,96,201
QE 850 DATA 34,208,2,169,39,96
       ,165,20,24,101,21,201
RD 860 DATA 39,176,10,32,180,1
       92,32,212,193,32,212,19
       2
JR 870 DATA 96,32,202,192,76,1
       07,193,24,165,20,101,21
FS 880 DATA 201,39,240,7,176,9
       ,165,2,32,127,192,32
CH 890 DATA 180,192,96,165,2,3
       2,127,192,32,202,192,96
FE 900 DATA 32,207,255,96,160,
       0,145,195,230,195,208,2
JF 910 DATA 230,196,96,96,162,
       1,32,198,255,165,55,133
GD 920 DATA 195,165,56,133,196
       ,169,0,141,238,194,169,
       255
XD 930 DATA 141,68,192,32,152,
       193,32,152,193,32,218,1
       93
```

```
X8 940 DATA 169,6,133,158,169,
       39,133,159,32,212,192,9
       6
8E 950 DATA 169,0,32,156,193,9
       6,169,246,133,253,169,1
       93
HR 960 DATA 133,254,160,0,177,
       253,201,255,208,1,96,32
FQ 970 DATA 156,193,230,253,20
       8,240,230,254,208,236,3
       9,8
ED 980 DATA 10,0,153,34,147,14
       4,34,59,58,151,53,51
XS 990 DATA 50,56,49,44,49,53,
       58,151,53,51,50,56
XE 1000 DATA 48,44,48,58,153,1
       99,40,49,52,41,59,0
KJ 1010 DATA 51,8,20,0,129,73,
       178,49,164,50,51,0
MA 1020 DATA 59,8,30,0,135,65,
       36,0,78,8,40,0
EP 1030 DATA 139,65,36,178,34,
       29,157,34,167,141,56,4
       8
GP 1040 DATA 58,128,0,86,8,50,
       0,153,65,36,0,92
SE 1050 DATA 8,60,0,130,0,104,
       8,70,0,141,56,48
PD 1060 DATA 58,137,50,48,0,14
       4,8,80,0,153,166,48
SD 1070 DATA 56,41,34,17,32,18
       ,80,82,69,83,83,32
RE 1080 DATA 65,32,75,69,89,32
       ,84,79,32,67,79,78
BK 1090 DATA 84,73,78,85,69,14
       6,34,59,0,162,8,90
SS 1100 DATA 0,161,66,36,58,13
       9,66,36,178,34,34,167
CH 1110 DATA 57,48,0,174,8,100
       ,0,153,34,147,34,59
JR 1120 DATA 58,142,0,255,32,2
       04,255,173,196,194,73,
       128
MF 1130 DATA 141,196,194,162,1
       8,160,8,24,32,240,255,
       162
XQ 1140 DATA 0,189,196,194,240
       ,26,32,210,255,232,208
       ,245
AA 1150 DATA 18,80,82,79,67,69
       ,83,83,73,78,71,32
AQ 1160 DATA 66,76,79,67,75,35
       ,32,0,238,238,194,169
CM 1170 DATA 0,174,238,194,32,
       205,189,169,46,32,210,
       255
AS 1180 DATA 169,146,32,210,25
       5,96,0,32,26,193,32,20
       4
DQ 1190 DATA 255,96,32,253,174
       ,32,158,173,160,0,177,
       71
JE 1200 DATA 72,200,177,71,170
       ,104,142,12,195,76,13,
       195
EG 1210 DATA 0,174,12,195,160,
       0,24,32,240,255,173,12
PG 1220 DATA 195,160,0,162,40,
       201,24,208,2,200,202,1
       69
XS 1230 DATA 32,32,210,255,202
       ,208,248,136,240,5,238
       ,12
HG 1240 DATA 195,208,227,162,0
       ,160,0,24,32,240,255,1
       62
HC 1250 DATA 0,181,217,9,128,1
       49,217,232,224,24,208,
       245
AG 1260 DATA 96,6,169,1,32,195
       ,255,169,36,141,215,19
       5
GQ 1270 DATA 169,48,141,216,19
       5,169,1,162,8,160,0,32
KD 1280 DATA 186,255,169,2,162
       ,215,160,195,32,189,25
       5,32
CQ 1290 DATA 192,255,169,64,32
       ,144,255,162,1,32,198,
       255
EJ 1300 DATA 32,144,255,32,207
       ,255,32,207,255,32,207
       ,255
RF 1310 DATA 32,207,255,201,0,
       240,67,32,204,255,32,2
       28
PS 1320 DATA 255,201,32,208,6,
       32,217,195,76,161,195,
       201
GD 1330 DATA 13,208,2,240,45,1
       62,1,32,198,255,32,207
P8 1340 DATA 255,168,32,207,25
       5,72,152,170,104,32,20
       5,189
JC 1350 DATA 169,32,32,210,255
       ,32,207,255,201,0,208,
       8
SR 1360 DATA 169,13,32,210,255
       ,76,129,195,32,210,255
       ,76
SJ 1370 DATA 185,195,169,1,32,
       195,255,32,204,255,96,
       36
DQ 1380 DATA 48,32,228,255,201
       ,32,208,1,96,201,13,20
       8
SE 1390 DATA 244,104,104,76,20
       6,195
```

©

COMPUTERS

ATARI

65XE	Call
130XE	139.95
520ST (RGB)	Call
520ST (MONO)	Call

COMMODORE

C-128	275.95

PRINTERS

STAR MICRONICS

SG-10	210.95
SG-15	369.00
SD-10	321.95
SD-15	441.00
SR-10	469.00
SR-15	582.00
SD-10	565.00
Powertype	299.95

LEGEND

808	154.95
1080	205.95
1380	259.95
1385	295.00

OKIDATA

Okimate 10	170.95
Okimate 20	210.00
182	219.95
192	349.95

PANASONIC

KX-P1080	209.00
KX-P1091	231.95
KX-P1092	369.00
KX-P1592	Call
KX-P1595	Call
KX-P3131	259.95
KX-P3151	425.00

CITIZEN

MSP10	259.00
MSP15	359.95
MSP20	335.95
MSP25	499.95

SEIKOSHA

SP-1000(C-64)	179.95
SP-1000(Centronics)	185.95

EPSON

Call for current pricing on all Epson models

PRINTER RIBBONS AND DUST COVERS AVAILABLE

SOFTWARE

BATTERIES INCLUDED

Homepak	31.95
Paper Clip	36.95

BRODERBUND

Bank Street Writer	32.95
Karateka	16.95
Lode Runner	18.95
Print Shop	27.95
Print Shop Companion	Call
Graphics Libraries I, II, & III ea	16.95

CONTINENTAL

Tax Advantage	34.95
Home Accountant	32.95

MICROPROSE

F-15 Strike Eagle	20.50
Silent Service	20.50
Kennedy Approach	20.50

OSS

MAC 65	48.95
Action	48.95
Basic XL	38.95
Basic XE	48.95
Tool Kits	18.95

SUBLOGIC

Flight Simulator II	31.95
Jet	Call
Night Mission Pinball	20.50

SYNAPSE

Synfile	31.95
Syncalc	31.95

DISK DRIVES

ATARI

1050	149.95
Happy 1050	299.95
Happy Enhancer	139.95
U S Doublers	54.95
DT Duplicator	149.95
DT Doubler	54.95
Indus GT	198.00
SF314	215.95
SF354	169.95
Haba 10 Meg Hard	Call

COMMODORE

1571	249.95
Enhancer 2000	185.95
Indus GT	205.00

INTERFACES

ATARI

MPP1150	45.95
U-Print/Port	49.95
U-Print/16K	69.95
U-Print/64K	79.95
PR Connection	64.95

COMMODORE

Super G	59.95
MW350	56.95
Xelec Super	64.95
G-Wiz	48.95
Microstuffer	69.95

DISKETTES

PRECISION	5¼	3½
SS/DD	$8.50	—
DS/DD	11.75	—

MAXELL		
MD1	$15.95	32.95
MD2	19.95	36.95

NASHUA		
SS/DD	—	29.95
DS/DD	—	32.95

All Diskettes Carry a Lifetime Warranty

INNOVATIVE CONCEPTS

F-N-F 25/Lock	12.50
F-N-F 50	12.50
F-N-F 50/Lock	16.50
Disk Notchers	3.95

MONITORS

TEKNIKA

MJ-10	189.95
MJ-22	254.95

ZENITH

ZVM 122	74.95
ZVM 123	74.95

AMDEK

300G	117.00
300A	127.00
310A	145.00
Color 300	175.95
Color 600	289.95
Color 700	469.00
Color 710	539.00

ATARI

SM124	174.95
SC1224	335.95

NEC

1201	139.00
1205	139.00
1260	79.95

SAKATA

SC100	159.00

THOMSON

CM365	289.95

14" RGB Color Composite Amber & Green Switch

CM366	189.95

14" Color Composite

MODEMS

Team Modem 100% Hayes Compatible	199.00

ATARI

XM301	39.95
Mpp1000E	52.95
MPP1200	199.95
OMI1200ST	185.95

300/1200 100% Hayes Compatible

COMMODORE

1670	169.95
Mitey Mo	59.95
Messenger V.I.P /Compuserve	46.95
Volks 6470	159.95

PAPER

WHITE 20 LB

2500 Shts. Laz. Edge	26.95
1000 Shts. Laz. Edge	16.95
500 Shts. Laz. Edge	11.95

ASSORTED PASTELS

2500 Shts. Laz. Edge	44.95
100 Shts. Laz. Edge	26.95
500 Shts. Laz. Edge	16.95
Mailing Labels 10000QTY	9.95

ORDER TOLL FREE 1-800-351-3442

CUSTOMER SERVICE AND PA RESIDENTS CALL 1-717-322-7700

Place orders Mon-Fri 9am-7pm. Customer service calls taken Mon-Fri 10am-4pm. No deposit on C.O.D. orders. Free freight on all prepaid cash orders over $300 in the continental U.S.A. APO & FPO orders add $5 per hundred. Priority mail add $10 per hundred. All foreign orders add 8%. Inter-shipping. UPS shipping add $4 per hundred East Coast, $5 per hundred West Coast. PA residents add 6% sales tax. Free shipping for PA residents. Orders by company and personal checks held 3 weeks. Defective products require prior return authorization. Defective products will be replaced or repaired according to warranty. No used or reconditioned products sold. Prices and availability subject to change without notice.

"Where Prices are Born, Not Raised."

WHITE HOUSE COMPUTER

P.O. Box 4025
Williamsport, PA
17701

VISA 4%. MASTER CARD 4%. AMERICAN EXPRESS 5%

Apple ProDOS Protector

Jason Coleman

These programs protect your Apple II ProDOS disks against unauthorized use by other people. If you're using DOS 3.3, see the "Guardian Angel" article elsewhere in this issue for a similar protection method.

"Apple ProDOS Protector" lets you protect any ProDOS disk from unauthorized use by others. Three files are required to make this system work. Before getting started, type in and save Programs 1–3 listed below, which are all written in Applesoft BASIC.

To begin the protection process, select the disk you want to protect, then load and run Program 1, "File Creator." The program asks you to enter a unique access code for the soon-to-be-protected disk. The access code can be any length and can contain any combination of letters, numbers, and symbols except for the comma and colon. Be sure to write the access code down for later reference—you may find it difficult or impossible to use the disk without it.

The program then creates a machine language file on disk named START.END.ML. You don't need a copy of Program 1 on the disk to be protected, only a copy of the START.END.ML file created by Program 1.

Next, you are asked to enter the name you wish to use for this disk's startup file. Make a note of this filename as well.

When Program 1 is finished, load Program 2 and save it on the disk to be protected, *using the filename you selected for the startup file.* Then load Program 3 and save it on the disk to be protected, too, *using the filename ENDUP.* The disk should now contain these three files:

1. START.END.ML, the machine language file created by Program 1.
2. Program 2, saved with the filename you selected for the startup file using Program 1.
3. Program 3, saved with the filename ENDUP.

This disk is now protected against most users. Only programmers proficient at working with the ProDOS machine language interface (MLI) can gain access without knowing the access code.

Using Protected Disks

When a protected disk is booted, the user is asked to enter the correct access code. If the access code is correct, the user is not allowed to use the disk. Anyone who doesn't know the code will not be able to break out of the program by pressing CTRL-C or CTRL-RESET.

When you are finished using a protected disk, load and run the ENDUP program (Program 3) to disable the CATALOG command so other users can't see what's on your disk.

Of course, no protection scheme is foolproof. But you should find this method sufficient to deter most casual users from accessing your ProDOS disks.

For instructions on entering these listings, please refer to "COMPUTE!'s Guide to Typing in Programs" in this issue of COMPUTE!.

Program 1: File Creator

```
11 100 HGR : HGR2 : TEXT : HOME
C5 110 FOR I = 8192 TO 8225: REA
       D J: POKE I,J: NEXT I
BD 120 OATA 32,0,191,128,28,32,1
       76,249,173,37,64,240,1,96
       ,169,22,141,37,64,32,0,19
       1,129,28,32,176,249,96,3,
       96,0,64,2,0
F4 125 POKE 8200,96: CALL 8192:A
        = PEEK (16421): POKE 820
       0,173: POKE 8207,A
41 130 INPUT "ENTER THE ACCESS C
       ODE WHICH YOU WILL USE TO
        ENTER YOUR OISK:";AC$
14 140 POKE 8226, LEN (AC$): FOR
        I = 8227 TO 8226 + LEN (
       AC$): POKE I, ASC ( MIO$
       (AC$,I - 8226)): NEXT I
75 150 PRINT CHR$ (4)"BSAVE STAR
       T.ENO.ML,A$2000,E";I
AE 160 HOME : PRINT "ENTER A FIL
       ENAME FOR YOUR STARTUP FI
       LE (NO LONGER THAN SEVEN
       LETTERS)": INPUT SF$: IF
       LEN (SF$) > 7 THEN 160
B4 170 PRINT CHR$ (4)"BLOAO BAS1
       C.SYSTEM,TSYS,A$2000"
5D 175 IF PEEK (8192) = 76 THEN
       185
DC 180 POKE 8677, LEN (SF$): FOR
        I = 8678 TO 8677 + LEN (
       SF$): POKE 1, ASC ( MIO$
       (SF$,I - 8677)): NEXT
B3 184 GOTO 190
42 185 POKE 8198, LEN (SF$): FOR
        I = 8199 TO 8198 + LEN (
       SF$): POKE I, ASC ( MID$
       (SF$,I - 8198)): NEXT
B1 190 PRINT CHR$ (4)"UNLOCK BAS
       IC.SYSTEM"
85 200 PRINT CHR$ (4)"BSAVE BASI
       C.SYSTEM,TSYS,A$2000"
BB 210 PRINT CHR$ (4)"LOCK BASIC
       .SYSTEM"
AB 220 NEW
```

Program 2: BOOTUP

```
07 100 ONERR GOTO 1000
75 102 RS = PEEK (1012): POKE 10
       12,RS + 10
25 105 HGR : HGR2 : TEXT : HOME
9B 110 PRINT CHR$ (4)"BLOAO STAR
       T.ENO.ML"
BC 120 L = PEEK (8226)
C1 130 FOR I = 1 TO L:CO$ = CO$
       + CHR$ ( PEEK (8226 + I))
       : NEXT
F2 140 INPUT "ACCESS COOE:";AC$
C3 150 IF AC$ < > CO$ THEN PR# 6
15 160 POKE 1012,RS
BD 165 A = PEEK (48944): POKE 82
       21,A
18 170 CALL 8192: HOME : NEW
7A 1000 RESUME
```

Program 3: ENDUP

```
11 100 HGR : HGR2 : TEXT : HOME
9B 110 PRINT CHR$ (4)"BLOAO STAR
       T.ENO.ML"
B3 115 A = PEEK (48944): POKE 82
       21,A
37 120 POKE 8200,96: CALL 8192
1D 130 A = PEEK (16421): POKE 82
       07,A: POKE 8200,173
3B 140 PRINT CHR$ (4)"BSAVE STAR
       T.ENO.ML"
0E 150 POKE 8207,0
34 160 CALL 8206
B1 170 NEW
```

©

HOTWARE: Software Best Sellers

This Month	Last Month	Title	Publisher	Remarks	Systems: Apple	Atari	Commodore	IBM	Macintosh
Entertainment									
1.		*Elite*	Firebird Licensees, Inc.	Action/adventure	•		•		
2.	3.	*Ultima IV*	Origin Systems, Inc.	Fantasy game	•	•	•		
3.		*Jet*	SubLogic	Jet simulation			•		
4.		*The Bard's Tale*	Electronic Arts	Adventure/role-playing game	•		•		
5.		*Temple of Apshai Trilogy*	Epyx	Fantasy/role-playing game		•	•		
Education									
1.	2.	*Math Blaster!*	Davidson	Introductory math program, ages 6-12	•	•	•	•	
2.		*Music Construction Set*	Electronic Arts	Music composition program	•	•	•		
3.	4.	*Color Me:The Computer Coloring Kit*	Mindscape	Children's artistic tool	•		•		
4.	1.	*Typing Tutor III*	Simon & Schuster	Typing instruction program	•		•	•	•
5.	5.	*New Improved MasterType*	Scarborough	Typing instruction program	•	•	•	•	
Home Management									
1.	2.	*Print Shop*	Broderbund	Do-it-yourself print shop	•	•	•		
2.	3.	*Printmaster*	Unison World	Do-it-yourself print shop	•	•	•	•	
3.		*Better Working Spreadsheet*	Spinnaker	Spreadsheet	•		•		
4	1.	*The Newsroom*	Springboard	Do-it-yourself newspaper	•		•	•	•
5.		*The Newsroom: Clip Art Collection, Vol. 1*	Springboard	Additional graphics	•		•	•	•

Copyright 1986 by Billboard Publications, Inc. Compiled by the Billboard Research Department and reprinted by permission. Data as of 6/7/86 (entertainment) and 6/14/86 (education and home management).

GREAT PRODUCTS FOR YOUR COMMODORE

promenade C1™

The Eprom Programmer. Thoughtfully designed, carefully constructed, the *promenade* C1™ is respected around the world for quality and value. The original software controlled programmer does away with personality modules and switches. Intelligent programming capability can cut programming time by 95%! With Disk Software ..... still just $99.50

CAPTURE™

Take control of your '64 or '128* with this easy to use cartridge. Lets you make a back-up disk of your memory-resident software. Your program is then fully accessible to you and your program can be re-booted from your disk 3-5 times faster. Or make an auto-starting cartridge using the *promenade* C1 and a CPR cartridge kit. Its magic!

CAPTURE™ is a bargain at 39.95

CARTRIDGE MATERIALS:

CPR-3 - Three socket board, case and 3 eproms, for use with **CAPTURE™** 29.95
PCC2 - Two 2732 (4K) or 2764 (8K) eproms. For '64 or '128 in 64 mode 4.95
PCC4 - Four sockets for 2764, 27128 or 27256 (32K) eproms. Bank switching 17.95
PCC8 - Like the PCC4 but twice the capacity. For multiple programs 29.95
PRB4 - Four sockets, eprom & battery backed RAM combination 24.95
PTM2 - Basic 128 mode cartridge board. Two 2764 or 27128 eproms ..... 5.95†
PTM4 - Four sockets, 27128 & 27256 eproms. 128 mode bank switcher ..... 19.95†
PCCH2 - Plastic case for above cartridges (except PCC8) 2.25
Eproms - Always in stock at competitive prices.

†available June '86.
*when in 64 mode.

EPROM ERASERS:

Datarase - Hobbyist eprom eraser, 2 at a time, 3 to 10 minutes 34.95
PE14 - Industrial quality eraser, 7 to 9 at a time 79.95
Starter Set - **CAPTURE™**, *promenade* C1 and one CPR3 kit 149.95
Deluxe Set - **CAPTURE™**, *promenade* C1, Datarase and two CPR3 kits 199.95

SHIPPING & HANDLING: USA - UPS SURFACE $3.00 FOREIGN (AIR MAIL ONLY) $13.00

JASON-RANHEIM
1805 INDUSTRIAL DRIVE
AUBURN, CA USA 95603

TO ORDER TOLL FREE 800-421-7731
FROM CALIFORNIA 800-421-7748
TECHNICAL SUPPORT 916-823-3284
FROM OUTSIDE USA 916-823-3285
MC, VISA, AMEX WELCOME

Commodore 128 Machine Language

Part 1

Jim Butterfield, Associate Editor

This article launches a new series on machine language programming for the Commodore 128. In this installment, we'll examine some basic architectural features of the 128, including memory banking, and look at a program that passes information between BASIC and ML.

The Commodore 128 is truly three computers in one—a Commodore 128 when in 128 mode, a Commodore 64 when in 64 mode, and a Z80-based CP/M computer when in CP/M mode. This series of articles discusses programming the computer in machine language in 128 mode.

When in this mode, the 128's 8502 microprocessor can execute the same instructions as the Commodore 64's 6510 microprocessor, and many of the programming techniques used on the 64 work exactly the same on the 128. These articles are directed especially at programmers who need to make the transition from 64 machine language to 128 ML programming. Of course, if you're familiar with 6502/6510 programming, but the 128 is your first Commodore computer, you can still benefit from the information presented here.

Ground Rules

Here are two simple ground rules to keep you out of trouble on the 128:

First, it's important to stay in bank 15 when writing programs with the computer's built-in machine language monitor (we'll explain what a bank is in a moment). This rule is necessary because of the 128's memory architecture, which can be confusing to a beginner. If you choose a bank number lower than 12, you may end up in a machine configuration which has no Read Only Memory (ROM), making it impossible for your program to call any of the computer's built-in ROM routines.

Second, stay away from areas of Random Access Memory (RAM) which are usually safe on the 64. On the 64, for instance, the cassette buffer located at 828–1019 ($033C–$03FB) is a good place to put short ML programs, and the free RAM block from 49152–53247 ($C000–$CFFF) is ideal for longer programs. Both areas are unusable on the 128, as you'll quickly learn if you try to put ML code there. The lower area contains critical system vectors and subroutines; if you change their contents, the system will crash. The higher area is covered by Kernal ROM; you can't easily put an ML program there and still have access to ROM routines.

Instead, the 128 has safe areas from 2816–3071 ($0B00–$0BFF) and 4864–7167 ($1300–$1BFF). The first area is the 128's cassette buffer, and the second area is currently unused by the system. In later articles, we'll provide more details on these rules as well as some exceptions to them.

Why Bank 15?

The 128 is capable of seeing its memory as 16 different banks numbered 0–15. The term *banks* is somewhat misleading, since a bank does not represent a separate 64K block of memory. Instead, each bank represents a different configuration or arrangement of the various available RAM and ROM elements. The bank number determines what the 128 sees within various areas. In some banks, the 128 sees nothing but RAM; in others it sees a combination of RAM and ROM; still other configurations include RAM, ROM, and input/output (I/O) addresses, and so on.

In fact, there are 256 possible memory configurations. Most of these, however, are of little or no use. For example, though you can configure the computer to see only half of its BASIC ROM and none of its Kernal ROM, it's hard to imagine any use for such an arrangement. Commodore has chosen 16 configurations which seem most useful, named the different configurations *banks*, and identified them with numbers from 0–15.

Figure 1 shows the configuration for bank 15. From locations $0002–$3FFF there is RAM. The 128 in the computer's name means that the computer has a total of 128K of RAM, which is arranged in two 64K blocks called *RAM 0* and *RAM 1*. Don't confuse these blocks with banks—some RAM from one or both of these blocks appears in every bank, but the amount varies.

The RAM in bank 15 is from RAM 0, the block that holds BASIC program text along with various buffers, vectors, and system variables and subroutines. More about the rest later. For the moment, it's important to notice that a BASIC program's working values—variables, arrays, and strings—are *not* contained in the same bank as the program text itself.

As shown by Figure 1, most addresses above 16384 ($4000) are seen as ROM. The BASIC interpreter alone occupies a hefty 28K, all the way up to 45055 ($AFFF). Above that, we have the machine language monitor and operating system (Kernal) interspersed with some I/O addresses and a tiny area earmarked for the memory management unit (MMU).

In the I/O section, from 53248–57343 ($D000–$DFFF), all the chips from the Commodore 64 appear in the same addresses. Thus, your favorite 64 POKEs to make sound effects and so forth work exactly the same in 128 mode. There are numerous extra I/O locations to do new jobs, such as controlling the 80-column video chip and reading the extra keys on the 128's keyboard.

At this point, we won't worry about the machinations of the MMU; it's enough to learn that bank 15 provides access to all the I/O chips as well as the Kernal ROM.

When you put a machine language program in RAM 0, you might be tempted to issue a BANK 0 statement from BASIC before you start the program with SYS. After all, bank 0 gives you access to all the memory in RAM 0. Don't do this: It's better to stay in bank 15.

Figure 2 shows the bank 0 configuration. Putting the computer in this configuration will certainly allow it to see your ML program in RAM 0. But the computer can't see its I/O chips or Kernal ROM. The computer has lots of memory, but no way to communicate with the outside world.

What's the lesson? Stay in bank 15. You are limited to 16K of RAM, but that's plenty for most applications. Later in this series, we'll discuss access to other configurations.

If you don't specify a bank, the computer defaults to bank 15. However, it's prudent to execute a BANK 15 statement just before any SYS from BASIC. This ensures that your program will work even if some other program has left the machine configured for a different bank. As a courtesy to other programmers (and users in general), programs that use other configurations should end by returning the machine to the default bank.

Memory Use In RAM 0

Figure 3 illustrates typical memory usage in the first 16K of RAM 0. Note that there are several unused memory areas available for program storage. Unless you're using a graphics mode, BASIC program space begins at 7168 ($1C00). (While programming in ML, you might want to avoid using an otherwise handy program known as the DOS Shell; it moves the start of BASIC up to $5B01 and occupies memory above $1A00—memory you may want to use for your own purposes.)

Figure 3 also reveals other unused or little-used memory zones. If you don't need to use a tape drive,

Figure 1: Bank 15

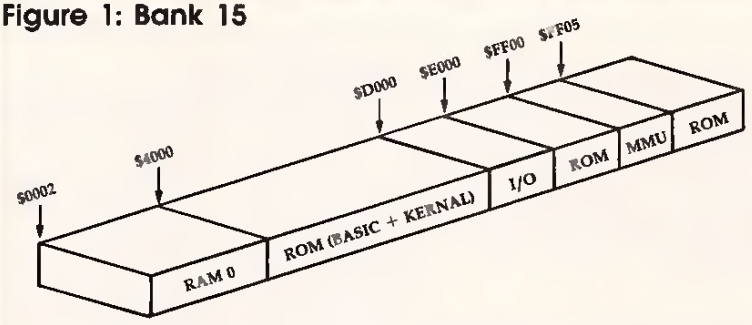


Figure 2: Bank 0

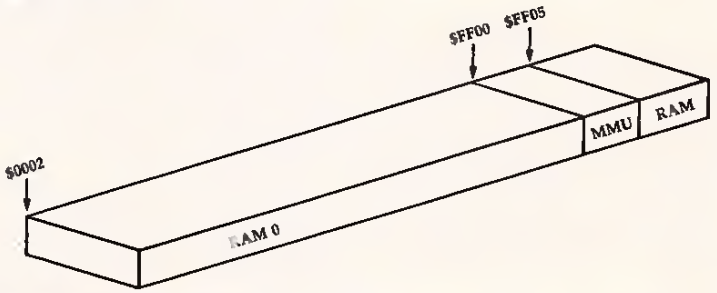
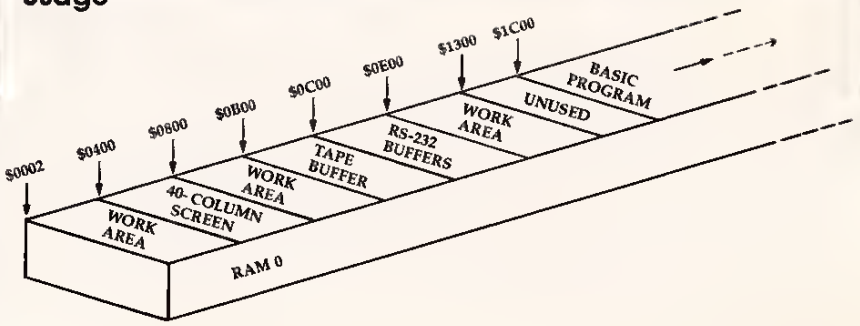


Figure 3: RAM 0 Memory Usage



the cassette buffer from 2816–3071 ($0B00–$0BFF) is free. If you aren't using telecommunications, the RS-232 buffers from 3072–3583 ($0C00–$0DFF) are also available. And there's a large block of empty memory marked *reserved for applications software* that stretches from 4864–7167 ($1300–$1BFF), providing over 2K of contiguous free space.

Friendlier BASIC

BASIC 7.0, the vastly improved BASIC in 128 mode, has several features that simplify the process of combining BASIC and ML. We won't explain all of them in detail, but here is a brief survey. (Your *System Guide* contains additional information.)

In addition to calling an ML routine, the SYS statement can also pass values from BASIC to ML. The values must be in the range 0–255 and are placed in the microprocessor's registers just before the ML routine takes over. Simply tack them onto the end of the SYS command, separated by commas. Conversely, the RREG command lets you read the processor's registers from BASIC after an ML routine has finished.

The BLOAD command can bring in any ML module (or a graphics screen, etc.) with no fuss or bother. The file loads into the same memory area from which it was saved, and BASIC continues with the next command. This is much simpler than the gyrations required in earlier versions of Commodore BASIC.

BASIC 7.0 also makes it easy to convert numbers between decimal and hexadecimal. The DEC function converts a hexadecimal string into a decimal number. The HEX$ function converts a decimal number into a hexadecimal string.

A Rudimentary Example

The following program isn't particularly useful, but may interest you in the 128's new features. It counts the number of 1 bits in any eight-bit number and prints them out in a table. You may not be excited to learn that the number 14 (binary 00001110) contains three 1 bits, while the number 16 (binary 00010000) contains only one, but the program does demonstrate how to pass information from BASIC to machine language and back again. We'll explain the purpose of each program line as we go. Here's the first one:

```
100 BANK 15
```

This statement puts the computer into bank 15, the safest configuration. Since the ML part of our program won't use any Kernal routines or I/O chips, we could use bank 0. But there's no advantage in doing so, and another time we might not be so lucky. Remember, it's always wise to set the bank explicitly rather than assume everyone's computer will be in bank 15.

```
110 DATA 162,0,74,144,1,232
    ,168,208,249,96
```

This is the short ML program, stored in the form of DATA statements. It takes a value from the accumulator (A register), counts the 1 bits in the value, and places the result in the X register.

```
120 FOR J=2816 TO 2825
```

The actual ML code goes in locations 2816–2825 ($0B00–$0B09), the bottom of the cassette buffer.

```
130 READ X:T=T+X
140 POKE J,X
150 NEXT J
```

Before the ML can be used, it has to be READ from the DATA line and POKEd into memory. A simple additive checksum detects most typing errors.

```
160 IF T<>1334 THEN STOP
```

If the program stops at line 160, you've made a typing error, most likely in the DATA statements. If not, the ML code is safely planted in memory and we can proceed to the job of bit counting.

```
200 FOR J=0 TO 20
```

We're going to count the 1 bits in numbers from 0–20. You can examine higher numbers if you like, but don't try anything over 255.

```
210 SYS 2816,J
```

This statement calls the ML program at its starting address of 2816 and passes the value of the variable J to the processor's A register. When the machine language program begins to run, the A register will contain that value. We could also have passed values to the X and Y registers, but this program doesn't require them.

```
220 RREG S,T
```

When we reach line 220, the ML program has returned control to BASIC. We'd like to know what values were in the processor's registers, expecially the X register, which contained the bit count. The RREG command reads the registers and places their values into BASIC variables. The A register goes into variable S and the X register goes into T. Now T contains the bit count.

```
230 PRINT J,T
240 NEXT J
```

That's all it takes. We print the value of J and the bit count T, then go back to do it again.

Yet To Come

We haven't touched yet on the 128's excellent built-in machine language monitor, nor have we explained how to "break the bank"—free ourselves of some of the constraining features of working within banks. Later in this series, we'll do all of this and more. ©

Copyright 1986 by Jim Butterfield

DISKETTES
BASF maxell.
5¼" SS $9 BOX OF 10 $10 BOX OF 10
5¼" DS $11 BOX OF 10 $15 BOX OF 10
5¼" HD $24 BOX OF 10 $33 BOX OF 10
3½" SS $8 BOX OF 5 $19 BOX OF 10
3½" DS $15 BOX OF 5 $35 BOX OF 10
Minimum Diskette Order $100
300/1200 MODEM
Hayes Compatible
Volksmodem 12 $159
Signalman Express...$199
Volksmodem 300 .....$55
RACORE
PCjr LIVES
NOW WITH DMA
• 2nd 360K, 10MB Or 20MB Drives
• Memory Plus Upgrades
• Keyboard Cables
• Serial Cables
FREE Catalog of Racore and other PCjr products
This Ad Supercedes All Previous Ads. IBM is a registered trademark of International Business Machines
MICRO MARKETING 404-441-1081 GA 800-652-9289 USA
875 GLEN RIDGE DR• LILBURN, GA 30247

Foolproof Input For Amiga BASIC

Tom Bunker

Here's an extremely handy tool for Amiga BASIC programmers—a routine that creates edit field boxes for accepting various kinds of keyboard input. The routine also demonstrates how well-designed subprograms can, in effect, add new commands to Amiga BASIC.

Amiga BASIC's ability to use custom subprograms is one of its most valuable features: It allows programmers to accumulate a library of very useful routines that can be attached to virtually any BASIC program. The simple requester window subprogram presented in the March 1986 issue of COMPUTE! is just one example. Another subprogram that should be in every programmer's collection is a foolproof input routine.

The ideal input routine would simulate the Amiga operating system's own edit field boxes. An example of such an edit field appears when you select the *Save as* option in Amiga BASIC's *Project* menu. A similar routine in BASIC would give your programs much more control than provided by the standard INPUT statement. It would be helpful, for instance, to be able to limit the number of characters that can be entered, or to limit numeric input to integers rather than print error messages after the fact. The input routine shown here has all of these capabilities and more.

Edit Fields In BASIC

The complete input routine consists of two subprograms: "Getline," which gets a line of input from the keyboard, and "Box," which Getline calls to draw an edit field box and cursor on the screen. The Box subprogram is very useful in its own right and can be used independently of Getline.

Getline lets you create the equivalent of an edit field box in Amiga BASIC. Here are some of its features:

- The main program which calls Getline sets the maximum length of input allowed.
- The Box subprogram draws an edit field box of appropriate size.
- The cursor inside the box can be flashing or nonflashing.
- The main program can select the type of input allowed: alphanumeric characters, real numbers, or integers.
- The range of alphanumeric characters accepted for input can be adjusted.
- Pressing the ESCape key aborts the input operation.
- A single keystroke can erase all input within the edit field box.
- The main program can display a default entry within the edit field box which the user can edit.

Getline can be used any time your program needs to accept input from the keyboard, for entry of data, filenames, or whatever. To use Getline, your program should first print any desired prompt message and leave the cursor at the point on the screen where input is to begin. Then you must call Getline using this general format:

CALL Getline (*string$,maxlength%, inputtype%*)

The string variable *string$* holds whatever default text you want to display inside the edit field box for the user to edit, and also returns the input entered by the user. For instance, if Getline is called as part of a save-data-to-disk routine, you could suggest a default filename or use a filename which the user has previously indicated. If you don't want to display anything within the edit field box when it appears, set this string variable to a null string ("") before calling Getline. In any case, Getline returns the user's input in this string variable after the subprogram passes control back to your main program.

The second parameter (*maxlength%*) is an integer which sets the maximum input length. For instance, if you want to limit input to 30 characters, you'd specify a 30 for this parameter by supplying either an integer variable or a constant.

The last parameter (*inputtype%*) is an integer which tells Getline which type of input to accept. There are three possible values:

0 accepts all alphanumeric characters without restriction.

1 accepts real numbers—the digits 0 to 9 and the decimal point.

2 accepts integers—only the digits 0 to 9.

The real and integer types also accept the plus and minus signs, but only in the first character position. Getline simply ignores all keystrokes that do not conform to the type of input selected.

CALLing Getline

Here are a couple of examples. Let's say you want the user to enter his or her name, up to 14 characters long, and you want your program to store the information in the string variable NAME$. The proper CALL would be

```
CALL Getline (NAME$,14,0)
```

If you want the user to enter a three-digit integer number (perhaps a telephone area code), the proper CALL would be

```
CALL Getline (NUMBER$,3,2)
```

Note that Getline always returns the user's input in a string variable. If the input you're seeking is an integer or real number, you can convert it from string to numeric form with the VAL function after Getline returns control to your main program.

Remember, too, that Amiga BASIC's CALL statement has an alternate syntax: You can omit the CALL keyword if you delete the parentheses surrounding the arguments. The following statements work the same as the examples above:

```
Getline NAME$,14,0
Getline NUMBER$,3,2
```

This syntax saves a bit of program space, but also sacrifices a certain amount of program clarity. If you include the CALL keyword, it is always clear to others that the program is calling a subprogram.

Special Keystrokes

When called, the Getline subprogram first draws an edit field box the proper size to hold the input. If the string variable supplied in the call is not a null string (two quotes with nothing between them), the subprogram prints the string inside the box. A flashing cursor indicates that the program is awaiting keyboard input. Like the Amiga operating system's own edit fields, Getline recognizes the following special keystrokes:

- ESCape exits the edit field and leaves the string variable with the value it had when Getline was called.
- RETURN exits the edit field and assigns the user's entry to the string variable.
- BACKSPACE deletes the character to the left of the cursor.
- DEL deletes the entry currently in the edit field.
- CURSOR LEFT moves the cursor one space to the left.
- CURSOR RIGHT moves the cursor one space to the right.

The last four commands, of course, are valid only if at least one character is within the edit field.

Customizing Getline

Note that Getline is designed to work only when Amiga BASIC's default font is used and Preferences is set to 80 columns. If you're using a 60-column screen or a different font, the text doesn't appear properly within the edit field box. You can modify the subprograms to solve this problem if you don't regularly use the default 80-column font.

If you don't want to bother with three parameters every time you call Getline, you can omit either the maximum string length or input type or both, as long as you also delete the corresponding items from the parameter list of the SUB statement. The Getline call can be made as simple as this:

```
Getline NAME$
```

In this case, the SUB statement would have to be changed to look for only one argument:

```
SUB Getline(inputstring$) STATIC
```

Getline substitutes default values for maxlength% or inputtype% when they are missing from the parameter list. Maxlength% defaults to 40, and inputtype% defaults to 0 (thus accepting all types of input). You can change these defaults too, if you wish.

Two variables in the Getline subprogram—*asc.low* and *asc.high*—determine the ASCII range of characters that are accepted in the edit field. You can change these variables to make the subprogram accept any range of characters desired, even to the extent of restricting input to only one key. They could also be declared in a SHARED statement and set by your main program.

The ESCape key aborts the input and exits the edit field. If your main program needs to know whether or not the edit field was terminated by ESCape (as opposed to a RETURN with no other input), add the following line to the Getline subprogram immediately following the SUB statement:

```
SHARED K
```

After the subprogram ends, your main program can test the value of K. If K=27, the ESCape key was pressed.

You can also program one or more of the special function keys to work in a similiar fashion by adding additional lines directly below the ESCape key line to test for any other ASCII value. For example, the addition of this line:

```
IF K>=129 AND K<=138 THEN EXIT
  SUB
```

makes all the function keys abort the input like ESCape. Your main program could then test to see if K is equal to the ASCII value of any of the function keys and take whatever action is desired.

By deleting a single line as instructed by comments within the subprogram, Getline will always start with an empty string. Other comments show how the flashing cursor can be changed to a nonflashing cursor and how the box around the edit field can be eliminated. To make these changes, it's not necessary to actually delete the lines which are indicated. Simply insert a REM at the beginning of the line to disable it; this has the same effect and is more easily reversed.

The Box Subprogram

To draw the box around the edit field, Getline calls the Box subprogram. This subprogram selects a rectangular area of the screen and alters it in one of four ways. You may find this technique useful for other purposes as well. Here is the general format of the Box subprogram call:

```
CALL Box (wide%,high%,border%
  ,mode%)
```

or

```
Box wide%,high%,border%,mode%
```

The first two parameters (*wide%* and *high%*) set the size of the boxed area by specifying the width and height in number of characters. The third parameter

(*border%*) changes the size selected by increasing or decreasing the area on all four sides by the number of pixels specified. If this argument is 0, the perimeter of the area falls on the character boundaries. The last parameter (*mode%*) can range from 0 to 3:

0 fills the box interior using a PATTERN statement.

1 inverts the interior of the box.

2 outlines the area using the foreground color.

3 fills the box interior using the foreground color.

The Box subprogram can be very useful when you want to erase a word or clear any rectangular section of the screen. Consider this statement:

```
COLOR background#:Box 30,1,0,3
  :COLOR foreground#
```

This erases a section of the screen 30 characters long without affecting any surrounding text. It sets the foreground color equal to the background color, fills the area, and resets the color. Of course, you can achieve the same effect by printing spaces, but the Box subprogram works much faster.

Getline Input Routine

Note: The left-arrow symbols in this listing indicate when to press **RETURN** at the end of each program line. Do not attempt to type the arrows themselves.

```
SUB Getline(inputstring$, maxlen
gth%, type%) STATIC←
'Value of type% should be Ø for
character, 1 for real, 2 for int
eger←
'Set default maximum length:←
defaultlength=4Ø←
IF maxlength%=Ø THEN maxlength%=
defaultlength ←
y=CSRLIN:x=POS(o):a$=""←
asc.low=32:asc.high=125 'Set ASC
II limits←
'Delete next line to disable edi
t mode:←
a$=inputstring$ ←
cursor=LEN(a$):strlength=LEN(a$)
←
'Delete next line to eliminate i
nput box:←
Box maxlength%,1,2,2 ←
Print.line:←
LOCATE y,x:PRINT a$+SPACE$(maxle
ngth%-LEN(a$))←
Getkey:←
k$=INKEY$ ←
'Delete next line for nonflashin
g cursor:←
count=count-1 ←
IF count<=Ø AND cursor<maxlength
% THEN←
LOCATE y,x+cursor:Box 1,1,Ø,1←
count=1ØØ 'Set cursor flash rate
:←
END IF←
IF k$="" THEN Getkey←
k=ASC(k$):count=Ø←
IF k=13 THEN inputstring$=a$:GOT
O Done 'Return key←
IF k=27 THEN Done 'ESCape key←
IF k>=asc.low AND k<=asc.high AN
D strlength<maxlength% THEN←
IF type%>Ø THEN 'Check if real o
r integer←
IF k<43 OR k>57 OR k=44 OR k=47
THEN Print.line←
IF (k=43 OR k=45) AND cursor>Ø T
HEN Print.line←
IF type%>1 AND k=46 THEN Print.l
ine←
END IF ←
LOCATE y,x+cursor:cursor=cursor+
1:strlength=strlength+1 ←
a$=LEFT$(a$,cursor-1)+k$+MID$(a$
,cursor)←
PRINT MID$(a$,cursor):GOTO Getke
y←
END IF←
IF k=31 AND cursor>Ø THEN 'Curso
r left←
cursor=cursor-1 ←
ELSEIF k=3Ø AND cursor<strlength
THEN 'Cursor right←
cursor=cursor+1←
ELSEIF k=127 THEN 'Delete entry←
a$="":cursor=Ø:strlength=Ø ←
ELSEIF k=8 AND cursor>Ø THEN 'Ba
ckspace key←
cursor=cursor-1:strlength=strlen
gth-1←
a$=LEFT$(a$,cursor)+MID$(a$,curs
or+2)←
END IF←
GOTO Print.line←
Done:←
LOCATE y,x←
PRINT inputstring$+SPACE$(maxlen
gth%-LEN(inputstring$))←
END SUB←
←
SUB Box(wide%, high%, border%, m
ode%) STATIC←
'wide% and high% set size expres
sed as number of characters←
'border% is to be given as numbe
r of pixels←
'mode% - use Ø for pattern fill;
1 to invert area←
'mode% - use 2 for area outline;
3 to fill area with foreground c
olor←
y=CSRLIN*8-9-border%:y1=y:IF y1<
Ø THEN y1=Ø←
x=POS(o)*8-9-border%:x1=x:IF x1<
Ø THEN x1=Ø←
x2=x+wide%*8+1+2*border% ←
IF x2>=WINDOW(2) THEN x2=WINDOW(
2)-1 ←
y2=y+high%*8+1+2*border% ←
IF y2>=WINDOW(3) THEN y2=WINDOW(
3)-1←
IF x1>x2 THEN x1=x2 ←
IF y1>y2 THEN y1=y2←
IF mode%=2 THEN LINE (x1,y1)-(x2
,y2),,b:EXIT SUB←
IF mode%=3 THEN LINE (x1,y1)-(x2
,y2),,bf:EXIT SUB←
AREA (x1,y1):AREA (x2,y1):AREA (
x2,y2):AREA (x1,y2)←
AREAFILL mode%←
END SUB←
←
```

©

Copies of articles from this publication are now available from the UMI Article Clearinghouse.

For more information about the Clearinghouse, please fill out and mail back the coupon below.

UMI Article Clearinghouse

Yes! I would like to know more about UMI Article Clearinghouse. I am interested in electronic ordering through the following system(s):

☐ DIALOG/Dialorder ☐ ITT Dialcom
☐ OnTyme ☐ OCLC ILL Subsystem

☐ Other (please specify)_____
☐ I am interested in sending my order by mail.
☐ Please send me your current catalog and user instructions for the system(s) I checked above.

Name_____
Title_____
Institution/Company_____
_____
Department_____
Address_____
City_____State_____Zip_____
Phone (_____)_____

Mail to: University Microfilms International
300 North Zeeb Road, Box 91 Ann Arbor, MI 48106

The Screen Machine II

Part 2
Pull-Down Menus In IBM BASIC

Charles Brannon, Program Editor

Last month we presented "The Screen Machine II," a full-featured drawing program for the IBM PC and PCjr. Pull-down menus make it quick and easy to use. Many programmers would like to add user interface tools like pull-down menus to their own programs, so this month we'll take a look at the techniques used in Screen Machine II. The programs require an IBM PC with color/graphics adapter and BASICA or a PCjr with Cartridge BASIC. A joystick or graphics tablet is optional but recommended.

"The Screen Machine II" is a powerful graphics program that lets you draw in full color with a complete set of drawing tools. It is designed to be as easy to learn as possible without encumbering advanced users. Last month in Part 1, we listed Screen Machine II without REMarks for the sake of brevity. This month, we're publishing the fully commented version with an explanation of how you can use the menu subroutines in your own programs. See Part 1 for an explanation of how to use Screen Machine II.

Why A Visual Interface?

The visual user interface—consisting of pull-down menus, icons, and screen windows—is rapidly becoming the most popular way to operate a personal computer. Since the Apple Macintosh was introduced in 1984, nearly a million Macs have been sold. The basic principles have been adapted by the Atari ST series and Commodore Amiga, and several similar shells are available for IBM PC computers—including Digital Research's *GEM,* IBM's *Topview,* and Microsoft's *Windows.* Even the older eight-bit computers, such as the Commodore 64, are being updated with visually oriented operating systems like *GEOS.*

Those who prefer this style sometimes say that the best advantage of the visual interface is that it makes you feel as if you have a tangible presence within the computer. Instead of viewing yourself as a somewhat remote user of the machine, an operator at a terminal, you feel more like a part of the system. Your sense of flow is enhanced because you can instantly recognize graphic metaphors (such as a picture of a disk) or simply scan through pull-down menus to see what commands are available and appropriate.

A drawing program that takes advantage of this approach lets you preview the figures you're drawing before you actually set them in stone (or phosphor). For instance, using a mouse controller, joystick, or graphics tablet, you can move the pointing arrow across the screen canvas, then click a button to set one endpoint of a line. Now, as you move the pointer, a "rubberband" line is drawn between the first point and the current cursor position. You can move the line around, changing its orientation and length, until you've put it right where you want it. Then you press a button again to stamp it down. Of course, if this still isn't what you want, an Undo command could restore the screen to its former state.

If you've never had a chance to work with pull-down menus, you might not appreciate their advantages. Since the menus let you both view and execute the program's commands, they serve two functions: They provide a way to use the program while acting as built-in documentation. Menus that drop down from the top of the screen let you work with nearly the full screen area, instead of cluttering it up with help screens or conventional menus.

On the other hand, if you prefer a written approach to communication, you may find the act of scurrying around a dynamic screen to be clumsy and inefficient, particularly if you have little trouble memorizing lots of commands and typing at least 30 words per minute. A program that seeks to keep everybody happy can provide alternative keyboard commands as well as menus and icons.

Programming Menus

Writing a program which incorporates a visual user interface can be tricky. The newest Microsoft BASICs—such as Microsoft BASIC for the Macintosh and Amiga BASIC—have built-in commands to create and manage pull-down menus. Creating a menu is as simple as listing the text in a series of MENU statements. There are even ON MENU GOSUB statements which set up *event traps* (BASIC interrupts) to detect menu selections. Other commands, such as ON MOUSE GOSUB, let the program read the pointing device and respond to button clicks.

IBM BASIC lacks these features, but does include event-trapping statements like ON STRIG GOSUB for the joystick. This makes it possible to simulate the operations which are handled automatically by the newer BASICs. When the user clicks the selection button on the pointing device, the program has to check to see if the pointer is within the menu bar (the first line of the screen). If so, it then checks to see if the arrow is pointing at one of the menu titles. If so, the program drops down the menu (saving the screen contents of the area overwritten by the menu box), and again checks the pointer position to see which menu item is being pointed at. The program highlights the item, and then unhighlights it if the pointer moves away from the item. Finally, when an item is selected (or when the menu selection is canceled), the program has to remove the menu from the screen, restoring the screen contents overlapped by the menu.

Again, all of these details are handled for the programmer in Macintosh and Amiga BASIC. Nevertheless, with enough programming, you can do the same thing in IBM BASIC. The key is being able to drop down a menu and then later restore the part of the screen overlapped by the menu.

BASIC's GET and PUT commands are the solution: GET is used to copy a rectangular portion of the screen into a storage array, and PUT copies the image from the array back to the screen. Naturally, this technique requires using a graphics mode, since you can't GET or PUT with the text screen. However, with a machine language routine to buffer part of the text screen, this method could be adapted for use with a text-only display adapter.

Simulated MENU Commands

Screen Machine II demonstrates how this technique works. It contains several subroutines which simulate the MENU commands and event traps found in Macintosh and Amiga BASIC. Fortunately, you don't have to know about the inner workings of these subroutines to use them in your programs. There are a few variables and arrays that need to be defined (some of these are initialized automatically), but you need only three GOSUBs to do everything:

GOSUB 11000 is used to add a menu title or menu item to the list of menus.

GOSUB 14000, used within a loop, tracks the arrow pointer and continually checks for a menu selection. If a menu is selected, it automatically handles the mechanics of dropping down the menu, getting a selection, and then restoring the screen. You then examine the variables MNID (menu id) and MNIT (menu item) to see which, if any, menu item was selected.

GOSUB 20000 reads the pointer position and optionally tracks the cursor automatically.

Essentially, these subroutines are substitutes for the MENU command, MENU function, and MOUSE function built into Macintosh and Amiga BASIC. Therefore, they can be very handy for translating Macintosh and Amiga programs into IBM BASIC.

A few other useful subroutines let you turn the cursor on or off and print text on the graphics screen in reverse-video. All of these routines let you set variables to allow special options or fetch additional information. Most importantly, they are designed to be used with any program, not just Screen Machine II, so you can easily add them to an existing program or use them as a starting point for your next project.

Screen Machine II is far too large to cover in detail, but the listing below (Program 1) is liberally commented with REMs. By following these comments you can easily deduce the flow of the program. If you didn't type in the program last month, you can enter this listing and omit the comments without ill effect. (Aside from the remarks, this month's program is identical to last month's.) In fact, the remarks take up too much memory to allow the program to run. If you type in the program as listed, use Program 2, "REMover," to remove all the remarks to create a runnable version.

REMover can be used to strip comments from other IBM BASIC programs, too. When you run REMover, first enter the name of the program you're deleting the REMs from, followed by a unique filename for the REMless program to be created. You then have two options. Option 1 changes all REM statements into a single apostrophe (the abbreviation for REM). This preserves the line in case it is the target of a GOTO or GOSUB (not a problem with Screen Machine II), but deletes the text of the remark. Option 2 deletes all REM or apostrophe statements, and if the REM is the only statement in the line, deletes the entire line as well. It's not safe to use Option 2 on programs that may branch to a line beginning with a remark, but it works just fine with Screen Machine II. Be sure you keep a copy of your unprocessed, remarked program for future reference.

Using Menus In Your Program

You can detach the menu package from the rest of the program either by deleting everything except lines 10000–21040, or by saving just the menu lines to disk as an ASCII file suitable for merging with another program. Just enter

```
LIST 10000-21040,"MENU.PAK"
```

to create an ASCII file on disk called MENU.PAK. You can then type MERGE "MENU.PAK" to add these lines to an existing program. If you are starting from scratch, type LOAD "MENU.PAK".

Before your program can call the menu package, you need to initialize certain variables. These variables are shown in lines 210–340 of

the Screen Machine II listing. See the section on GETXY below to see how to set ACC, DACC, FROZEN, XMAX, YMAX, XOFF, and YOFF. Check the section on CURSOR_ON and CURSOR_OFF for information on setting the CURSOR flag. Finally, you can choose sound effects by setting SNDFX to −1. If you set it to zero, no sound is used.

Lines 9000–9340 illustrate how to define your menu structure. For example, the DATA statements for the Picture menu are

```
DATA 1,0,1,"Picture   "
DATA 1,1,1,"Undo    U"
DATA 1,2,1,"New     ^N"
DATA 1,3,1,"Open    O"
DATA 1,4,1,"Save    S"
DATA 1,5,1,"View    V"
DATA 1,6,1,"Quit    ^Q"
```

The first number is the *menu-ID*, the number specifying which menu is being defined. It must be at least 1, and less than 9 (unless you change line 11000 to allow more than 8 menus and/or more than 8 items in each menu). The next number is the *menu item*. A menu item of 0 defines the title of the menu, and other numbers specify the name of each item within the menu. The next number is a *status flag* for that menu item. A value of 1 is normal. Use 2 to display a checkmark next to an item.

The Ghost In The Machine

For example, the Tools menu puts a checkmark next to the currently selected tool. This allows a menu to be used to select items, show which commands are available, *and* show the status of each menu item.

When you specify a value of 0 for the menu status flag, that menu item is *ghosted out*, or dimmed. A ghosted item is still readable, but the text is distorted, indicating to the user that this particular command is currently disabled or not appropriate at the current time. This helps users avoid confusion over what they can and cannot do in a given situation—they can always access a command unless it's ghosted out.

There are many times when a program would want to change these assignments, depending on program context. For instance, after you select a new tool, the previous tool is reset to a flag of 1 (normal), and the new item is set to 2 (checked). In the Preferences menu, some of the menu items—such as Bkgd Color—are ghosted out when you are in 640 × 200 mode (because you can't change the screen color in this mode), but revert to normal when you select another graphics mode.

Initializing Menus

Here are descriptions of all the major routines in Screen Machine II:

11000 MENU To initialize or change the value of a menu item, assign values to the variables MNID, MNIT, and MNSTR$, then GOSUB 11000. MNID holds the number of the menu (1–8); MNIT holds which menu item is being changed (0–8, where 0 is the menu title); and MNSTR$ is the text displayed as the menu title or menu item. A program can modify all of these items at any time, changing the appearance of the menu when it drops down.

The subroutine at line 9000 in Screen Machine II can be used as a model for initializing your own menus. This routine stores the values in the arrays MTITLE$, MFLAGS, and MITEMS. It stores the number of the highest menu-ID used so far in TOPID to find out how many menus there are. The one-dimensional array MITEMS holds the number of menu items in each menu. MTITLE$ and MFLAGS are two-dimensional arrays that use MNID and MNIT to point to the title string and flag setting for a menu item. Hence, MFLAGS(1,2) holds the status flag value of menu 1, item 2. MTITLE$(3,0) holds the title of the third menu.

It can be convenient to assign values to these arrays directly—for example, when you just want to change one menu item's status flag. MFLAGS(3,4)=0 would ghost out the fourth item of the third menu. You could change it back to normal with MFLAGS(3,4)=1. Or you might want to change the text of a menu entry by modifying the MTITLE$ array. For instance, a menu item could initially read SOUND ON, then change to SOUND OFF after you've turned on the sound. This is an alternative to using the checkmark, but it can be confusing. Does SOUND ON imply that the sound is already on, or that the item will turn on the sound? Most programs use checkmarks to avoid this confusion.

12000 MENU_REFRESH Use GOSUB 12000 to display the title bar of your menus after you've initialized them after successive calls to the subroutine at line 11000. Your program should try to avoid using the top line of the screen, but you can always use GOSUB 12000 to redisplay the menu bar if the top line is lost. This routine also links in the positions of each menu item so that the MENU_POLL routine (line 14000) can figure out which menu you are pointing at. These positions are stored in the MX array.

**13000 RVSMSG$** There is no easy way to print reverse-video text on the IBM graphics screen, but this is the effect we want when we highlight a menu title or menu entry. The menu bar is also printed in reverse. To display reverse text, set MSG$ to the text you'd like to PRINT, then GOSUB 13000. This routine prints the text, uses GET to copy the text into an array, then uses the PRESET option of PUT to stamp down a reverse copy of the text.

14000 MENU_POLL This is the workhorse of the menu package. When you call this routine, it checks to see if the pointer is pointing at a menu title and the button is pressed. If not, it just RETURNs, leaving the variables MNID and MNIT set to 0. Otherwise, it drops down the menu, gets the selection, and exits with MNID and MNIT set to the value of the menu-ID and menu item. If the user canceled the selection by moving outside of the menu box, MNIT and MNID are reset to 0.

This routine uses simple sound effects as additional audio cues for the user. If you set the variable SNDFX to 0, you won't get sound effects. If you want them, set SNDFX to −1.

This routine also preserves your screen display and cursor position. If the keyboard is used for menu selection, the keyboard offset (see below) is increased to speed up movement between menu items.

Be aware that this routine works like INKEY$—if there is no menu selected yet, it immediately

RETURNs. You need to continually call this routine within a loop until MNIT is nonzero, meaning that a menu has been selected. The cursor arrow is updated automatically throughout the menu selection process. Even if no menu is selected, calling this routine continually calls the GETXY routine at line 20000 to update the cursor position.

15000 This subroutine is used only by MENU_POLL to flash a selected menu item.

16000 MENU_DOWN Given a value in MNID, this routine drops down the indicated menu, saving the screen contents erased by the menu in the MSAVE% array (initialized in line 11010). This routine is really only called by the MENU_POLL routine when a menu has been selected, but you may be able to use it for some special effects. To remove the menu, be sure to use the next routine, MENU_AWAY, to discard the menu and restore the screen contents.

17000 MENU_AWAY Again, this is really only used by MENU_POLL to roll away the menu after the user has made a selection. You can use it to remove the menu and restore the screen if you used MENU_DOWN to drop the menu yourself.

Cursor Routines

18000 CURSOR_ON

19000 CURSOR_OFF The arrow pointer is defined in this program in the subroutine at line 3000, used to select various graphics modes. You could excerpt line 3050 (as long as you remember to DIM ARROW% (32) at the start of your program) to use this cursor in your own program. Otherwise, draw your cursor on the screen and use GET (x1,y1)–(x2,y2),ARROW% to copy the cursor into the ARROW% array (x1,y1, and x2,y2 are opposite endpoints of an imaginary rectangle that should completely enclose the cursor shape). The GETXY routine (20000) needs to know the width and height of the cursor, so store these values in XARROW and YARROW.

The cursor is animated with the XOR option of PUT. When you PUT the arrow, it combines itself with the existing screen display so that it is always in contrast. Just think of the cursor as a stamp that uses "negative ink"—ink that reverses the color of anything it touches. For example, a white arrow on a black background would be white, but on a white background would be black. The magic of XOR is that when you PUT the shape back down on top of itself, it reverses the action, removing the arrow and restoring the previous screen contents. Although PUT with XOR can be flicker-prone, you can reduce the flicker by increasing the delay between drawing the arrow and erasing it.

You don't have to worry about updating the arrow cursor yourself. As long as you continually call either MENU_POLL (14000) or GETXY (20000), the arrow position is updated while the routine is checking the pointer position. But you have to remember to remove the arrow from the screen before you draw anything that might overlap the arrow. If you drew a white line through the cursor while it was resting on a white area, you've drawn a white line through the black arrow. When the arrow is PUT back on top of itself to erase the arrow, the conditions are no longer the same. The cursor reverses itself, so the cursor is gone, but you're left with a black line where the cursor used to be (remember the "negative ink" analogy).

Therefore, your program needs to erase the cursor from the screen before drawing anything. After you've drawn your figure, you can turn the cursor back on, or just allow GETXY (20000) to turn it back on automatically the next time you check for the cursor position.

So use GOSUB 19000 to turn off the cursor, and GOSUB 18000 to turn it back on. This is not the same as setting the CURSOR flag (see GETXY below). The CURSOR flag prevents or enables automatic cursor updates, but doesn't graphically affect the display. However, you should turn off the cursor with GOSUB 19000 before you turn off the cursor flag. If this seems confusing, examine the drawing routines in Screen Machine II (lines 1000–1660) to see how this is done.

20000 GETXY This routine is the core of the whole package. It is used any time a routine wants to know where the cursor is pointing. As part of the normal checks for the joystick position, it can also update the cursor automatically. To get automatic cursor tracking, be sure to set the CURSOR flag to −1; otherwise you are responsible for your own cursor movement. For use with a joystick or graphics tablet, this routine converts the joystick/tablet values to actual screen positions by multiplying the controller position times the values XRATIO# and YRATIO#.

XRATIO# and YRATIO# are the horizontal and vertical size of the screen divided by the maximum X and Y values of the controller (the lower-right position). When multiplied by the joystick value, these values scale the joystick values to actual screen coordinates. A range of 0–255 multiplied by 1.251 (319/255) gives us a range from 0–319.

Set XRATIO# to the horizontal size of the screen divided by the maximum value of the controller. If the maximum value of the joystick is 132, and you're working with the 320 × 200 mode, then XRATIO# =320/132. Similarly, YRATIO# is the number of rows divided by the maximum vertical position of the controller, as in YRATIO#=200/130.

Reading The Pointing Device

XOFF is the minimum horizontal value of the joystick, and YOFF is the smallest vertical value returned by the joystick. You can test this by pushing the joystick to the upper-left corner, then executing PRINT STICK(0), STICK(1). Similarly, you can move the joystick to the lower-right corner and PRINT STICK(0), STICK(1) to assign default values to XMAX and YMAX as shown in lines 230 and 240. Screen Machine II illustrates how to set these values in the screen setup routine at line 3000. Also, the Calibrate function from the Preferences menu (refer to lines 2440–2510) is used to read the values of XMAX, YMAX, XOFF, and YOFF.

XOFF and YOFF, the minimum (top-left) values of the controller, are used to adjust the calculations, as well as to check whether a stylus is pressed against a graphics tablet

surface. For example, the KoalaPad usually returns 7 and 7 as its X and Y values when there is no surface contact. This can be used as a convenient shortcut. While in drawing mode, for example, you start drawing by clicking the button, and stop drawing by either clicking the button again, or simply moving the stylus off the tablet surface.

Another note about the KoalaPad: It is extremely sensitive to glitches unless you bear down on the tablet with firm pressure. Unfortunately, pressing too hard will score the tablet surface. If you don't press hard enough, the values jitter uncontrollably. Fortunately, BASIC is too slow to notice most of these glitches, which occur for a fraction of a second before the values reset to normal. If you compile the program, though, it is much more sensitive to these glitches. An averaging routine could be used to detect the glitches and ignore them, but would greatly slow down the uncompiled program.

For keyboard control, GETXY allows the cursor keys to be used to move the cursor. If the keyboard was used instead of the controller, the variable KEYMODE is set to −1; otherwise KEYMODE is reset to 0 when the joystick or graphics tablet is used.

Cursor movement can be very slow, though, if you are moving only one pixel at a time. You must set the variable DACC to the number of pixels you'd like the pointer to move each time a cursor key is pressed, and initialize the variable ACC to this value. If the key is pressed successively or held down until it repeats, ACC counts up, accelerating the speed of the arrow cursor. When the key is released or a different key is pressed, ACC is reset to the value of DACC.

On the other hand, if DACC is a negative quantity, no acceleration is performed. Every keypress just advances the cursor by the absolute value of DACC (as if it were positive). You can change these values throughout your program depending on the context. The MENU_POLL routine sets DACC to −8 during menu selection so that the cursor keys move by one screen line at a time without accelerating.

Reading The Keyboard

If the flag FROZEN equals −1, the joystick or graphics tablet is ignored in favor of the keyboard. Do this when you need keyboard control while the joystick is plugged in. Although the keyboard is always active, it attempts to increment or decrement the values of MX and MY, but these variables are continually reset to the scaled value of the joystick position. With the graphics tablet, we can tell if the stylus is pressed down and ignore the tablet position if it isn't. So the keyboard and tablet work interchangeably, but you need to set FROZEN to −1 if you want keyboard control only while ignoring the joystick.

Line 20050 checks for keyboard equivalents that indirectly activate menu entries. Most commands in Screen Machine II have keyboard equivalents—O for Open, L for Lines, CTRL-N (^N) for New, etc. In addition to streamlining the program for advanced users, keyboard commands satisfy those who are uncomfortable with pointing and clicking. If you don't mind memorizing every keystroke, you don't really need menus. However, not every menu item is always represented by a keystroke, and it's hard to find unique assignments for every menu item.

You really don't need to bother with keyboard equivalents, but if you want them, initialize the string CM$ as illustrated in line 9060. For each keyboard equivalent, include the keyboard character followed by the digit of the menu-ID and the digit of the menu item for that menu selection. This limits you to nine menus and items, but makes keyboard checking quick. INSTR$ is used to instantly find out if the command key is part of CM$, and just as easily retrieve the subsequent values of MNID and MNIT. Strictly speaking, this line does not really belong in GETXY, but we need it here to use the same keystroke that GETXY uses to check for a cursor key.

Study the program listing for more ideas. Since nearly every line is commented, it should be easy enough to follow. We would be interested in seeing the kinds of programs you develop using these techniques.

Quick Reference To Subroutines

12000 MENU_REFRESH
Uses MNID, MNIT, and MNSTR$ to initialize a menu item.
MNID: Which menu
MNIT: Which menu item
Fills the arrays MTITLE$( ), MFLAGS( ), MITEMS( )

13000 RVSMSG$
Diplays MSG$ in reverse video at current cursor position.

14000 MENU_POLL
If a menu item is found, returns menu-ID in MNID and menu item in MNIT; otherwise MNID=0 and MNIT=0.

18000 CURSOR_ON
If the cursor flag is set (CURSOR<>0), draws pointer cursor and tells the package that the cursor is on the screen by setting TOGGLE=1.

19000 CURSOR_OFF
If the cursor flag is set (CURSOR<>0), removes pointer cursor from screen and tells the package that the cursor is not on the screen by setting TOGGLE=0.

20000 GETXY
Polls keyboard and optionally the joystick (if FROZEN=0). See text for necessary initialization. Returns MX, MY, MB (mouse/joystick position and button status). If CURSOR flag is nonzero, automatically updates an arrow cursor at position MX, MY.

For instructions on entering these listings, please refer to "COMPUTEI's Guide to Typing In Programs" in this issue of COMPUTEI.

Program 1: The Screen Machine II

```
NP 100 'Screen Machine II
GB 110 'Requires CGA or PCjr, AB
       ASIC 2.x or Cartridge BAS
       IC
HA 120 DEFINT A-Z
CP 130 'Test for PCjr
PA 140 PCJR=0:ON ERROR GOTO 150:
       SOUND OFF:CLEAR ,,,32768'
       :DEFINT A-Z:PCJR=-1
JF 150 IF NOT PCJR THEN RESUME 1
       60
JJ 160 ON ERROR GOTO 0
HC 170 '
OL 180 'Constants used by menuin
       g package:
HG 190 '
MG 200 'To compile this program,
        remove apostrophe from f
       ollowing line, delete lin
       e 11010
QK 210 'OIM MTITLE$(8,8),MFLAGS(
       8,8),MITEMS(8),MSAVE%(160
       8),mx(8):TOP10=0
AF 220 OIM ARROW%(32),ZZTEMP%(64
       8) 'reserve memory for cu
       rsor, temp use
MJ 230 'XMAX=100:YMAX=100:XOFF=3
       :YOFF=3 'recommended for
       joystick.
FF 240 XMAX=250:YMAX=230:XOFF=7:
       YOFF=7 'recommended for u
       se with touch tablet
AH 250 HIGHLIGHT=2 '# of flashes
```

```
        when menu item selected
QD 260 TRUE=-I:CURSOR=TRUE 'enab
       les automatic arrow curso
       r
MP 270 KEY OFF:SCREEN 0,0,0:WIDT
       H 40:COLOR ,I,I:CLS:LOCAT
       E 4,II,0:COLOR I2:PRINT "
       SCREEN MACHINE II"
MP 280 LOCATE 7,I2:COLOR 10:PRIN
       T"CharIes Brannon"
KD 290 COLOR I4:LOCATE 13,10:PRI
       NT"One moment, pIease..."
KH 300 GOSUB 9000 'initialize th
       e menus
CK 310 SMDDE=I:COLR=I:GOSUB 3000
        'sets up screen, XRES,YR
       ES,MAXCOLOR,SWIDTH,ARROW%
        cursor
XH 320 SNDFX=TRUE 'set to non-ze
       ro for sound effects
IP 330 ACC=1:DACC=1 'DACC is #pi
       xels moved per keystroke.
        If negative, makes keybo
       ard movement constant, el
       se allows acceIeration
QD 340 FROZEN=0  'if frozen=true
        (-1), joystick or touch
       tabIet is ignored in favo
       r of keyboard input
HA 350 '
LH 360 'Program starts here
HE 370 '
FI 3B0 COLR=1:TOOL=1 'current co
       Ior, drawing action
QP 390 STRIG ON 'enable mouse bu
       tton
ID 400 MX=XRES/2:MY=YRES/2:NX=MX
       :NY=MY:GOSUB I8000 'turn
       on cursor initiaIly
JJ 4I0 DIM UNDO%(15000) 'buffer
       portion of screen
HL 420 '
FJ 430 'Main Ioop:
HP 440 '
EI 4S0 WHILE TRUE 'i.e. forever,
        if true=-1
BN 460  GOSUB 18000:MB=0:MNID=0
CM 470  WHILE MNIO=0 AND MB=0 'w
       hiIe there's no menu seIe
       ction and no button press
LF 4B0    GOSUB 14000 'poII menus
BH 490  WEND
GP 500  IF MB<>0 THEN GOSUB 1000
        'draw
BM 5I0  IF MNID THEN GOSUB 2000
       'process menus
EM 520 WEND
HD 530 '
QD I000 WHILE MB:GOSUB 20000:WEN
       O 'wait for button reIea
       se
NH I0I0 GOSUB 19000 'turn off cu
       rsor
AC 1020 IF MY>=CY THEN COLR=INT(
       MX/XR#):GOSUB 6000:RETUR
       N
EC I030 GET (I,B)-(XRES-2,CY-1),
       UNOD% 'save screen in un
       do buffer.
JC 1035 SCM$=CM$:CM$="" 'tempora
       riIy disable keyboard co
       mmands
LG 1040 ON TDDL GOSUB I070,1170,
       1300,I430,1560,1630
LN I045 CM$=SCM$ 'restore keyboa
       rd commands
JK I0S0 RETURN
NF 1060 'Drawing routine
FK 1070 IF PENUP AND NOT KEYMDDE
        THEN RETURN 'drawing on
       Iy works with pen down
DH 10B0 CURSOR=0 'disable cursor
       for faster drawing
GA 1090 WHILE MB=0 AND (NOT PENU
       P OR KEYMDDE)
ES 1100 SX=MX:SY=MY:GOSUB 20000:
       MY=-MY*(MY>7 AND MY<CY)-
       D*(MY<8)-(CY-1)*(MY>=CY)
LM 1110 LINE (SX,SY)-(MX,MY),COL
       R 'connect the Iine
FK 1120 WEND
PJ 1130 WHILE MB:GOSUB 20000:WEN
       D 'wait for button reIea
       se
HC 1140 CURSOR=TRUE
JM 1150 RETURN
QQ 1160 'Draw Iines
MN 1170 SX=MX:SY=MY:CURSOR=0 'di
       sable cursor during Iine
        drawing
MP 1180 WHILE MB=0
LL 1190  LINE (SX,SY)-(MX,MY),0
       'erase previous Iine
QG 1200  GOSUB 20000:MY=-MY*(MY>
       7 AND MY<CY)-B*(MY<8)-(C
       Y-1)*(MY>=CY)
BL 1210  LINE (SX,SY)-(MX,MY),CO
       LR 'draw new Iine
EA 1220  EX=MX:EY=MY
FP 1230 WEND
PD 1240 WHILE MB:GOSUB 20000:WEN
       D 'wait for button reIea
       se
CL 12S0 PUT (1,8),UNDO%,PSET 're
       store mangIed screen
ED 1260 LINE (SX,SY)-(EX,EY),COL
       R 'draw finaI Iine
IN 1270 CURSOR=TRUE
JH 12B0 RETURN
FL 1290 'Draw boxes
CF 1300 SX=MX:SY=MY:CURSOR=0 'di
       sable cursor
LO 1310 WHILE MB=0
DF 1320  LINE (SX,SY)-(MX,MY),0,
       B 'erase previous box
PB 1330  GOSUB 20000:MY=-MY*(MY>
       7 AND MY<CY)-B*(MY<8)-(C
       Y-1)*(MY>=CY)
KD 1340  LINE (SX,SY)-(MX,MY),CO
       LR,B 'draw new box
FL 1350  EX=MX:EY=MY
GK 1360 WEND
QJ 1370 WHILE MB:GOSUB 20000:WEN
       D 'wait for button reIea
       se
CG 13B0 PUT (1,B),UNDD%,PSET 're
       store mangIed screen
LA 1390 LINE (SX,SY)-(EX,EY),COL
       R,8 'draw finaI Iine
HM I400 CURSDR=TRUE
IG 1410 RETURN
QB 1420 'Draw circIes
DA 1430 SX=MX:SY=MY:CURSOR=0 'di
       sable cursor
MJ 1440 WHILE MB=0
PM 14S0  CIRCLE (SX,SY),SOR(ABS(
       SX-MX)^2+ABS(SY-MY)^2),0
QM 1460  GOSUB 20000:MY=-MY*(MY>
       7 ANO MY<CY)-B*(MY<8)-(C
       Y-I)*(MY>=CY)
NP 1470  CIRCLE (SX,SY),SQR(ABS(
       SX-MX)^2+ABS(SY-MY)^2),C
       OLR
FG 1480  EX=MX:EY=MY
GF I490 WEND
PI 1S00 WHILE MB:GOSUB 20000:WEN
       D 'wait for button reIea
       se
FA 1510 GOSUB 3000:PUT (1,B),UND
       O%,PSET 'restore mangIed
        screen
JA 1S20 CIRCLE (SX,SY),SOR(ABS(S
       X-EX)^2+ABS(SY-EY)^2),CO
       LR
CL I530 CURSOR=TRUE:GOSUB 12000:
       GDSUB 6000 'redraw scree
        n bar and color bars in
       case circIe overwrote it
JB 1540 RETURN
QN 1550 'Spraycan
GB I560 WHILE MB=0 AND (NDT PENU
       P OR KEYMDDE)
JE 1570  GOSUB 20000:IF MY<12 OR
        MY>CY-S THEN 1590
BG 1580  GOSUB I9000:PSET (MX+4-
       8*RND,MY+4-8*RND),CDLR
GH I590 WEND
QI 1600 WHILE MB:GOSUB 20000:WEN
       D
JK 1610 RETURN
LO 1620 'Paint
NA 1630 ON ERRDR GOTO I660:PAINT
        (MX,MY),COLR:LINE (0,0)
       -(XRES-I,YRES-I),,B:GOSU
       B 6000:GOSUB 12000
FA I640 ON ERRDR GDTO 0:WHILE MB
       :GOSUB 20000:WEND 'relea
       se
JG I650 RETURN
KA I660 RESUME NEXT
BD 1670 'Menu handIer:
GF 2000 ON MNID GDSUB 2030,2320,
       2380 'Picture, Tools, Sc
       reen
IP 20I0 RETURN
QD 2020 'File menu
PC 2030 ON MNIT GDSUB 2060,2080,
       2100,2170,2240,2300 'Und
       o,New,Open,Save,View,Oui
       t
JI 2040 RETURN
JG 20S0 'Undo:
AF 2060 GOSUB 19000:PUT (1,8),UN
       DO%,PSET:RETURN
ED 2070 'New:
HE 20B0 GOSUB 3000:RETURN
IH 2090 'Open:
NB 2I00 TYP$="OPEN":GOSUB 4000 '
       get filename
QJ 21I0 IF FILENAME$="" THEN 213
       0
NC 2I20 ON ERRDR GOTO 5500:DEF S
       EG=SEGADR:BLDAO FILENAME
       $,0
JD 2130 ON ERROR GOTD 0:CLOSE#1
SD 2140 LINE (0,0)-(XRES-1,YRES-
       1),,B:GOSUB 12000:GOSUB
       6000
JN 2150 RETURN
GD 2160 'Save:
MK 2I70 TYP$="SAVE":GOSUB 4000 '
       get filename
NA 2I80 IF FILENAME$="" THEN 22I
       0
FN 2190 GET (1,B)-(XRES-2,CY-1),
       UNDO%:CLS:PUT (1,B),UNDO
       %,PSET
DL 2200 ON ERROR GOTO 5500:DEF S
       EG=SEGADR:BSAVE FILENAME
       $,0,SCRLEN'
KP 2210 ON ERROR GOTO 0:CLOSE#1:
       GOSUB 3000:PUT (I,8),UND
       D%,PSET
IG 2220 RETURN
NM 2230 'View:
DF 2240 GOSUB 19000:CURSOR=0
ED 22S0 GET (1,B)-(XRES-2,CY-1),
       UNDO%:CLS:PUT (I,8),UNDO
       %,PSET
HB 2260 WHILE MB=0:GOSUB 20000:W
       END
QG 2270 WHILE MB:GOSUB 20000:WEN
       D
MF 22B0 GOSUB 3000:PUT (1,B),UNO
       O%,PSET:CURSOR=-1:RETURN
KD 2290 'Quit
CM 2300 SCREEN 0,0,0,0:END 'use
       SYSTEM to exit to DDS
KK 2310 'Tools menu
```

```
KF 2320 MFLAGS(MNID,TDDL)=1 'tur
        n off previous tool
MC 2330 MFLAGS(MNID,MNIT)=2:TOOL
        =MNIT 'turn on current t
        ool
JD 2340 RETURN
HB 2350 STOP 'protect subroutine
        s from accidental execut
        ion
IE 2360 '
FJ 2370 'Screen:
NF 2380 IF MNIT<4 THEN SMODE=MNI
        T-2*(MNIT=3):GDSUB 3000
PK 2390 IF MNIT=4 THEN CDLDR ,1:
        MFLAGS(MNID,4)=2:MFLAGS(
        MNID,5)=1
BM 2400 IF MNIT=5 THEN CDLDR ,2:
        MFLAGS(MNID,4)=1:MFLAGS(
        MNID,5)=2
IF 2410 IF MNIT=6 THEN BG=(BG+1)
         AND 15:IF SMDDE=1 THEN
        CDLDR BG ELSE CDLDR ,BG
GF 2420 IF MNIT=7 THEN FRDZEN=ND
        T FRDZEN:MFLAGS(MNID,MNI
        T)=1-FRDZEN
QE 2430 IF MNIT<>8 THEN RETURN
GA 2440 GDSUB 19000:LDCATE 1,1:M
        SG$=LEFT$("Move stick to
         upper left, press butto
        n."+SPACE$(B0),SWIDTH):G
        DSUB 13000
IJ 2450 WHILE STRIG(1)=0:XDFF=ST
        ICK(0):YDFF=STICK(1):WEN
        D
NJ 2460 WHILE STRIG(1)<>0:WEND '
        wait for release
FO 2470 LDCATE 1,1:MSG$=LEFT$("M
        ove stick to lower right
        , press button."+SPACE$(
        B0),SWIDTH):GDSUB 13000
HK 24B0 WHILE STRIG(1)=0:XMAX=ST
        ICK(0):YMAX=STICK(1):WEN
        D
NC 2490 WHILE STRIG(1)<>0:WEND '
        wait for release
BH 2500 XRATID#=XRES/XMAX:YRATID
        #=YRES/YMAX
IF 2510 GDSUB 12000:RETURN
KA 2520 'Set up a screen, given
        XRES,YRES,PCJRMDDE
NG 3000 GDSUB 19000 'turn off cu
        rsor
KO 3010 IF SMDDE=PMDDE THEN 3030
KD 3020 DN SMDDE GDSUB 3110,3150
        ,3030,3030,3190
CP 3030 PMDDE=SMDDE
ND 3040 SWIDTH=INT(XRES/8):XRATI
        D#=XRES/XMAX:YRATID#=YRE
        S/YMAX 'screen width
PE 3050 CLS:PSET (10,10):DRAW "b
        m10,10d3e313f5":GET (10,
        10)-(17,17),ARRDW%  'cre
        ate cursor
ME 3060 XARRDW=8:YARRDW=8 'horiz
        ontal and vertical size
        of cursor
JD 3070 CLS:LINE (0,0)-(XRES-1,Y
        RES-1),,B 'border
PC 30B0 GDSUB 6000:GDSUB 12000
KI 3090 RETURN
HP 3100 '
DD 3110 SCREEN 1:COLOR 0,1:CDLR=
        1:XRES=320:YRES=200:BG=0
        :MAXCDLDR=4
HA 3120 GOSUB 3230:MFLAGS(3,1)=2
        :SEGADR=&HBB00:SCRLEN'=1
        63B4
PJ 3130 MFLAGS(3,4)=2:MFLAGS(3,5
        )=1:MFLAGS(3,6)=1
JL 3140 RETURN
QN 3150 SCREEN 2:XRES=640:YRES=2
        00:MAXCDLDR=2:CDLR=1
JG 3160 GOSUB 3230:MFLAGS(3,2)=2
        :SEGADR=&HB800:SCRLEN'=1
        6384
ID 3170 MFLAGS(3,4)=0:MFLAGS(3,5
        )=0:MFLAGS(3,6)=0
JH 3180 RETURN
JK 3190 SCREEN 5:XRES=320:YRES=2
        00:MAXCDLDR=16:CDLR=1
FE 3200 GOSUB 3230:MFLAGS(3,3)=2
        :SEGADR=&H1800:SCRLEN'=3
        2768'
KB 3210 MFLAGS(3,4)=0:MFLAGS(3,5
        )=0:MFLAGS(3,6)=1
IH 3220 RETURN
JP 3230 MFLAGS(3,1)=1:MFLAGS(3,2
        )=1:MFLAGS(3,3)=-PCJR:RE
        TURN 'reset modes
CM 3240 'Get a filename...
EG 4000 GDSUB 19000:GET (1,8)-(X
        RES-2,CY-1),UNDD% 'save
        screen
JG 4010 MSG1$="Please enter name
        ":MSG2$="of picture to "
        +TYP$
JP 4020 TW=SWIDTH/2-10:LINE (TW*
        8-10,50)-(TW*8+160,100),
        0,BF:LINE (TW*8-10,50)-(
        TW*8+160,100),,B:LINE (T
        W*8-8,52)-(TW*8+158,98),
        ,B 'draw box
HD 4030 LDCATE 8,SWIDTH/2-LEN(MS
        G1$)/2:PRINT MSG1$:LDCAT
        E 9,SWIDTH/2-LEN(MSG2$)/
        2:PRINT MSG2$
PG 4040 LINE (TW*8-5,78)-(TW*8+1
        55,89),,B:LDCATE 11,TW+1
        :MAXLEN=1B:GDSUB 5000
IP 4050 FILENAME$=EDT$:IF FILENA
        ME$>"" THEN IF MID$(EDT$
        ,LEN(EDT$)+3*(LEN(EDT$)>
        3),1)<>"." THEN FILENAME
        $=FILENAME$+".PI"+CHR$(48
        +SMDDE)
PK 4060 PUT (1,8),UNDD%,PSET 're
        store screen
JD 4070 RETURN
BM 4080 'Get a line of text (EOT
        $) of maximum length MAX
        LEN
QP 5000 EDT$="":IX=POS(0):IY=CSR
        LIN:XI=IX:KBD=-1:IF MAXL
        EN=0 THEN MAXLEN=79-IX
GB 5010 WHILE K8D<>13
GE 5020  XI=LEN(EDT$)+IX:LDCATE
        IY,XI:PRINT "_";:KBD$=IN
        PUT$(1)
PF 5030  KBD=ASC(KBD$):LDCATE IY
        ,XI:PRINT" ";
DC 5040  IF K8D=8 AND LEN(EDT$)>
        0 THEN EDT$=LEFT$(EDT$,L
        EN(EDT$)-1)
KH 5050  IF LEN(EDT$)<MAXLEN AND
         (K8D AND 127)>=32 THEN
        EDT$=EDT$+K8D$:LDCATE IY
        ,XI:PRINT K8D$;
GI 5060 WEND
JE 5070 RETURN
GB 50B0 'Error trap:
HL 5500 CLOSE #1 'close any file
FE 5510 GOSUB 19000:GET (1,8)-(X
        RES-2,CY-1),UNDD% 'save
        screen
KK 5520 TW=SWIDTH/2-10:LINE (TW*
        8-10,50)-(TW*8+160,100),
        0,BF:LINE (TW*8-10,50)-(
        TW*8+160,100),,B:LINE (T
        W*8-8,52)-(TW*8+158,98),
        ,B 'draw box
PK 5530 IF ERR>=52 THEN MSG1$="D
        OS ERROR #"+STR$(ERR):EL
        SE MSG1$="ERROR #"+STR$(
        ERR)+" in line"+STR$(ERL
        )
JF 5540 MSG2$="(R)etry or (C)anc
        el"
PD 5550 LDCATE 8,SWIDTH/2-LEN(MS
        G1$)/2:PRINT MSG1$:LDCAT
        E 10,SWIDTH/2-LEN(MSG2$)
        /2:PRINT MSG2$
BF 5560 K8D$=INPUT$(1):IF K8D$<>
        "r" AND K8D$<>"R" AND K8
        D$<>"c" AND K8D$<>"C" TH
        EN 5560
ND 5570 PUT (1,8),UNDD%,PSET 're
        draw screen
HL 5580 IF K8D$="r" DR K8D$="R"
        THEN RESUME ELSE RESUME
        NEXT
NN 5590 'Draw color bars
FH 6000 XR#=XRES/MAXCDLDR:CH=11:
        CY=YRES-CH-1
PH 6010 LINE (0,CY)-(XRES-1,YRES
        -1),0,8F
EJ 6020 FOR I=0 TO MAXCDLDR-1
DA 6030 LINE (I*XR#+2,CY+3)-(I*X
        R#+XR#-3,CY+CH-3),I,8F
QN 6040 NEXT
BH 6050 LINE (0,CY)-(XRES-1,YRES
        -1),,B
JD 6060 LINE (CDLR*XR#,CY+2)-(CD
        LR*XR#+XR#-1,CY+CH-2),,B
JF 6070 RETURN
FP 6080 'Initialize the menus
BK 9000 RESTORE 9090
GO 9010 WHILE MNSTR$<>"x"
NJ 9020  READ MNID,MNIT,MFLAG,MN
        STR$
ME 9030  IF MNSTR$<>"x" THEN GOS
        UB 11000
FG 9040 WEND
PI 9050 MFLAGS(3,3)=-PCJR 'allow
        /disallow special jr mod
        e
IP 9060 CM$="U11"+CHR$(14)+"12D1
        3S14V15"+CHR$(17)+"16D21
        L22R23C24A25P26B36K37J3B
        " 'key followed by corre
        sponding MNID and MNIT
KI 9070 RETURN
LJ 9080 'structure is MenuId,Men
        uItem,flag (0,1,2) and t
        itle for each entry
BD 9090 DATA 1,0,1,"Picture "
HE 9100 DATA 1,1,1,"Undo   U"
IN 9110 DATA 1,2,1,"New    ^N"
LH 9120 DATA 1,3,1,"Open   O"
PN 9130 DATA 1,4,1,"Save   S"
EN 9140 DATA 1,5,1,"View   V"
EK 9150 DATA 1,6,1,"Quit   ^Q"
IH 9160 '
ND 9170 DATA 2,0,1," Tools
        "
HJ 9180 DATA 2,1,2," Draw      D
        "
ML 9190 DATA 2,2,1," Line      L
        "
PH 9200 DATA 2,3,1," Rectangle R
        "
ID 9210 DATA 2,4,1," Circle    C
        "
JC 9220 DATA 2,5,1," Airbrush  A
        "
PI 9230 DATA 2,6,1," Paint     P
        "
ID 9240 '
GP 9250 DATA 3,0,1,"Preferences
        "
HG 9260 DATA 3,1,2," 320 x 200
        "
OI 9270 DATA 3,2,1," 640 x 200
        "
KO 9280 DATA 3,3,0," 320x200 PCj
        r"
DB 9290 DATA 3,4,2," cyn/mag/wht
        "
JI 9300 DATA 3,5,1," red/grn/yel
        "
```

```
JN 9310 DATA 3,6,1," Bkgd color
        B"
ED 9320 DATA 3,7,1," Keyboard
        K"
IB 9330 DATA 3,8,1," Calibrate
        J"
QD 9340 DATA ,,,x
HM 10000 'Menu package runs from
         lines 10000-29999
MC 10010 'Graphics adaptor requi
         red (works with a wide
         variety of modes)
FN 10020 'This entry point is fo
         r defining an individua
         l menu entry
DM 10030 '(equivalent to MENU me
         nuid,menuitem,flag,menu
         string$ in Mac/Amiga BA
         SIC)
HF 10040 'Pass variables MNID,MN
         IT,MFLAG, and MNAME$
MH 10050 'MNID=which menu
JH 10060 'MNIT=which item (or 0
         to define menu title)
PG 10070 'MFLAG=0 (ghosted), 1 (
         normal), 2 (checked)
DE 10080 'MNSTR$=name of menu or
          menu item
FN 10090 'You can also fill the
         MENUTITLE$(), MFLAGS(),
          MITEMS() arrays direct
         ly.  Set Topid to # of
         last menu, and call MEN
         U_REFRESH to update oth
         er arrays
FC 10100 'include the DIM statem
         ents following the MENU
          entry point
ID 10110 'If you don't set the v
         ariable SWIDTH (screen
         width) to 40 or 80, it
         defaults to 80
LN 10120 '
LH 10130 '*** MENU ***
LF 10140 '
HK 11000 MAXMENUS=8:MAXITEMS=8 '
         maximum # of menus and
         menu items
GF 11010 IF NDT MENUINIT THEN DI
         M MTITLE$(MAXMENUS,MAXI
         TEMS),MFLAGS(MAXMENUS,M
         AXITEMS),MITEMS(MAXMENU
         S),MSAVE%(800*MAXITEMS+
         8),MX(MAXMENUS):TDPID=0
         :MENUINIT=-1
JE 11020 IF MNID<1 DR MNID>MAXME
         NUS DR MNIT<0 OR MNIT>M
         AXITEMS THEN PRINT "ILL
         EGAL MENU PARAMETERS":S
         TDP
IH 11030 MTITLE$(MNIO,MNIT)=MNST
         R$:MFLAGS(MNID,MNIT)=MF
         LAG
OL 11040 IF MNIT>MITEMS(MNID) TH
         EN MITEMS(MNID)=MNIT
BA 11050 IF MNID>TOPID THEN TOPI
         D=MNID
IF 11060 RETURN
MA 11070 '
GD 11080 'Next entry point redra
         ws menu bar at top of s
         creen
NI 11090 '
HH 11100 '***  MENU_REFRESH ***
LL 11110 '
QJ 12000 IF SWIDTH=0 THEN IF XSI
         ZE THEN SWIDTH=INT(XSIZ
         E/8+.5) ELSE SWIOTH=80
DM 12010 MSG$=" ":MX(0)=8:SVX=PO
         S(0):SVY=CSRLIN
GL 12020 FOR MI=1 TO TOPID:MX(MI
         )=MX(MI-1)+8+LEN(MTITLE
         $(MI,0))*8:MSG$=MSG$+"
         "+MTITLE$(MI,0):NEXT:MS
         G$=MSG$+SPACE$(SWIDTH-L
         EN(MSG$))
KA 12030 LOCATE 1,1:GOSUB 13000
         'print MSG$ in reverse
         video
HC 12040 LOCATE SVY,SVX:RETURN
ML 12041 '
EF 12050 'Print MSG$ in reverse
         video
BI 13000 X1=POS(0)*8-8:Y1=CSRLIN
         *8-8:PRINT MSG$;:X2=X1+
         LEN(MSG$)*8-1:IF X2>=SW
         IDTH*8 THEN X2=SWIDTH*8
         -1
LL 13010 GET (X1,Y1)-(X2,Y1+7),Z
         ZTEMP%:PUT (X1,Y1),ZZTE
         MP%,PRESET:RETURN
LA 13020 '
CL 13030 ' Following entry point
          checks for menu select
         ion
DH 13040 'If a menu item is sele
         cted, returns MNID (men
         u id) and MNIT (menu it
         em)
MM 13050 '
EQ 13060 '*** MENU_POLL ***
ME 13070 '
MD 14000 XSAVE=POS(0):YSAVE=CSRL
         IN
LP 14010 MNIT=0:MNIO=0:GOSUB 200
         00 'get "mouse" coordin
         ates
BJ 14020 IF MY>7 OR MB=0 THEN RE
         TURN 'no menu event
IO 14030 WHILE MB:GOSUB 20000:WE
         ND 'wait for button rel
         ease
EH 14040 MI=1:WHILE MI<=TOPID AN
         D NOT (MX>=MX(MI-1) AND
          MX<=MX(MI)):MI=MI+1:WE
         ND
EL 14050 IF MI>TOPID THEN RETURN
          'no menu selected
DI 14060 MNIO=MI 'found which me
         nu, but not which item
HJ 14070 IF SNDFX THEN SOUND 100
         00,.5
FP 14080 GOSUB 16000:GOSUB 20000
          'drop menu, check "mou
         se"
KI 14090 SAVDACC=DACC:SAV$=CM$:C
         M$="":IF KEYMODE THEN M
         Y=2:NY=MY:DACC=-8 'spee
         d up keyboard temporari
         ly, to make menu select
         ion easier
DN 14100 WHILE MX>=MX(MNID-1) AN
         D MX<=MX(MNID) AND MB=0
          GOSUB 20000 'sets mx,m
         y,mb
MM 14110  MI=INT(MY/8):IF MI>MIT
         EMS(MNID) THEN GOTO 141
         50
AF 14130  IF MI=MNIT OR MFLAGS(M
         NID,MI)=0 THEN 14180
IJ 14140  GOSUB 19000 'turn curs
         or off during rendering
GC 14150  IF MNIT>0 THEN LOCATE
         MNIT+1,INT(MX(MNID-1)/8
         +2):PRINT MTITLE$(MNID,
         MNIT) 'un-reverse previ
         ous item
D4 14160  IF MI>0 AND MI<=MITEMS
         (MNID) THEN MNIT=MI:LOC
         ATE MNIT+1,INT(MX(MNID-
         1)/8)+2:MSG$=MTITLE$(MN
         ID,MNIT):GOSUB 13000:IF
          SNDFX THEN SOUND 20000
         ,.1 'print MSG$ in reve
         rse
CD 14170 IF MI>MITEMS(MNID) THEN
         MNIT=0
LL 14180 WEND
KE 14190 IF MX<MX(MNID-1) OR MX>
         MX(MNID) THEN MNIT=0 'm
         oved away from menu, so
          menu item is invalid
GD 14200 IF MNIT THEN GOSUB 1500
         0 'flash selection
LP 14210 GOSUB 17000 'remove men
         u
IA 14220 WHILE MB:GOSUB 20000:WE
         ND 'wait for button rel
         ease
MG 14230 'Delete next line if yo
         u want to know if a men
         u was selected, even if
          no menu item was selec
         ted.
HG 14240 IF MNIT=0 THEN MNID=0:I
         F SNDFX THEN SOUND 150,
         2:SOUND 50,1 'uh-oh
DG 14250 GOSUB 18000:DACC=SAVDAC
         C:CM$=SAV$:LOCATE YSAVE
         ,XSAVE 'restore main pr
         ogram's settings
JB 14260 RETURN
MM 14270 '
CL 14280 'Flash selected menu it
         em MNIT
JM 15000 IF MNIT=0 OR HIGHLIGHT=
         0 THEN RETURN
FM 15010 MSG$=MTITLE$(MNID,MNIT)
         :FOR MI=1 TO HIGHLIGHT:
         LOCATE MNIT+1,XP:GOSUB
         13000 'reverse it
DM 15020 IF SNDFX THEN SOUND 100
         00+MI*500,.1
HG 15030 LOCATE MNIT+1,XP:PRINT
         MSG$ 'unreverse it
BN 15040 NEXT:RETURN
MA 15050 '
CH 15060 'Drop down a menu, give
         n MNID.  Saves backgrou
         nd of image for when me
         nu is restored
HD 15070 '*** MENU_DDWN ***
NM 15080 '
KE 16000 WX1=MX(MNID-1):WX2=MX(M
         NID):WY1=8:WY2=8+8*MITE
         MS(MNID):XP=INT(WX1/8)+
         2
JC 16010 GOSUB 19000 'erase curs
         or
CL 16020 LDCATE 1,XP-1:PRINT " "
         +MTITLE$(MNID,0)
FP 16030 GET (WX1-2,WY1)-(WX2+2,
         WY2+2),MSAVE% 'save men
         u area
DG 16040 LINE (WX1-2,WY1-1)-(WX2
         +2,WY2+2),,8 'draw bord
         er
MH 16050 LINE (WX1-1,WY1)-(WX2+1
         ,WY2+1),0,8F  'erase in
         side of border
MM 16060 FOR MI=1 TO MITEMS(MNID
         )
JO 16070  LOCATE MI+1,XP:PRINT M
         TITLE$(MNID,MI)
DL 16080  IF MFLAGS(MNID,MI)=2 T
         HEN PSET (WX1,MI*8+5):D
         RAW "f2e5"
CA 16090  IF MFLAGS(MNID,MI)=0 T
         HEN GET (WX1,MI*8)-(WX1
         +LEN(MTITLE$(MNID,MI))*
         8+7,MI*8+7),ZZTEMP%:PUT
          (WX1,MI*8),ZZTEMP%,PSE
         T:PUT (WX1+1,MI*8),ZZTE
         MP%
QL 16100 NEXT MI
ID 16110 RETURN
MJ 16120 '
QP 16130 '*** MENU_AWAY ***
MB 16140 '
```

```
JA 17000 GDSUB 19000 'erase curs
         or
QK 17010 PUT (WX1-2,WY1),MSAVE%,
         PSET
AB 17020 LOCATE 1,XP-1:MSG$=" "+
         MT1TLE$(MN1D,0):GDSUB 1
         3000
IF 17030 RETURN
MA 17040 '
KN 17050 'Cursor ON
PN 1B000 IF CURSOR=0 DR TDGGLE=1
          THEN RETURN 'no cursor
         , or cursor already on
HJ 1B010 PUT (MX,MY),ARROW%:TDGG
         LE=1:RETURN
MJ 1B020 'Cursor DFF
GE 19000 IF CURSDR=0 DR TOGGLE=0
          THEN RETURN 'no cursor
         , or cursor already off
FK 19010 PUT (MX,MY),ARROW%:TDGG
         LE=0:RETURN
MM 19020 '
DP 19030 'Following routine chec
         ks "mouse", returns scr
         een coords MX,MY, and b
         utton status MB (0 if n
         ot pressed, else 1).
BA 19040 'Requires that XRATIO#
         and YRATID# set to rati
         o of horizontal/vertica
         l resolution divided by
          maximum value of joyst
         ick.  XOFF and YDFF are
          the lowest values retu
         rned by joystick or tab
         let.
BG 19050 'XRES and YRES are the
         width and height of the
          screen in pixels
HN 19060 'If CURSOR flag is set
         to -1, the routine upda
         tes an arrow cursor.
DH 19070 'If FRDZEN flag is non-
         zero, disables joystick
         /tablet.
EK 190B0 'Be sure to initialize
         ACC=1 and DACC.  ACC is
          an accelerating distan
         ce moved by cursor keys
         , reset to DACC to stop
          acceleration.  If DACC
          is negative, movement
         is constant, with no ac
         celeration.
MJ 19090 'Two flags returned are
          KEYMDDE (0 if joystick
         /pad was just used, els
         e -1) and PENUP (0 for
         no contact with pad or
         joystick at far upper-l
         eft corner, else -1).
LH 19100 '
KK 19110 '*** GETXY ****
MP 19120 '
DN 20000 M8=0:PENUP=0
JL 20010 IF NDT FRDZEN THEN S0=S
         TICK(0):S1=STICK(1):M8=
         STR1G(1):IF S0<>XDFF DR
          S1<>YDFF THEN NX=INT((
         S0-XDFF)*XRATIO#):NY=IN
         T((S1-YOFF)*YRATIO#):KE
         YMDDE=0:ELSE PENUP=-1
JN 20020 MK$=INKEY$:KY=0:IF MK$=
         "" THEN IF  TIMER>=TM!
         THEN ACC=ABS(DACC):TM!=
         TIMER+.1:GDTD 20060 ELS
         E 20060
LM 20025 KY=ASC(M1D$(MK$,2)+CHR$
         (0)):MB=MB DR -(KY=B2):
         KEYMDDE=-1
EE 20030 NX=-(NX+ACC*(KY=75)-ACC
         *(KY=77))*(KY<>71):NY=-
         (NY+ACC*(KY=72)-ACC*(KY
         =80))*(KY<>71)
MA 20040 IF KY=PK THEN ACC=ACC+2
         *(ACC<13)*(DACC>0):PK=K
         Y:ELSE ACC=ABS(DACC):PK
         =KY
DC 20050 KY=ASC(MK$):IF NOT (KY>
         47 AND KY<5B) THEN WHER
         E=INSTR(CM$,CHR$(KY+32*
         (KY>96 AND KY<123))):IF
          WHERE THEN MNID=VAL(MI
         D$(CM$,WHERE+1,1)):MNIT
         =VAL(M1D$(CM$,WHERE+2,1
         )):1F MFLAGS(MN1D,MN1T)
         =0 THEN MNIT=0:MNID=0 E
         LSE GDSUB 21010
HK 20060 IF NX=MX AND NY=MY THEN
          RETURN
DP 20065 XBDUND=XRES-XARRDW:YBOU
         ND=YRES-YARROW
EI 20070 NX=-NX*(NX>0 AND NX<=XB
         DUND)-XBDUND*(NX>XBDUND
         )-(NX<1)
FD 20080 NY=-NY*(NY>0 AND NY<=YB
         OUND)-YBDUND*(NY>YBOUND
         )-(NY<1)
PN 20090 GDSUB 19000:MX=NX:MY=NY
         :GOSUB 1B000
HP 20100 RETURN
JJ 21010 XP=1NT(MX(MN1D-1)/B)+2:
         MSG$=" "+MTITLE$(MNID,0
         ):GOSUB 19000
GG 21015 LDCATE 1,XP-1:PRINT MSG
         $
NJ 21020 IF SNDFX THEN SDUND 100
         00,.1
FL 21030 LOCATE 1,XP-1:GDSUB 130
         00
IQ 21040 RETURN
```

Program 2: REMover

```
CI 10 'REMover-deletes REMS from
       a program
AF 20 CLS:PRINT"REMover: Deletes
       REMs"
CL 30 PRINT:PRINT"Enter name of
       ASCII program to edit":LIN
       E INPUT ": ";ASCFILE$
MK 40 PR1NT:PRINT"Enter name of
       ASCII program to create":L
       INE INPUT ": ";CREATE$
DH 50 DPEN ASCFILE$ FOR INPUT AS
       #1
ND 60 DPEN CREATE$ FDR OUTPUT AS
       #2
PL 70 PRINT:PRINT"Level I change
       s REM statements to '"
FK B0 PRINT:PRINT"Level II delet
       es all REM lines and remov
       es REMs from end of line."
FD 90 PRINT:LINE INPUT "Level I
       (1) or II (2): ";LV$:LV=VA
       L(LV$):IF LV<1 DR LV>2 THE
       N 90
HK 100 WH1LE NOT EOF(1)
IB 110  LINE INPUT#1,L$:PRINT"*"
       ;
DL 120  RP=INSTR(L$,"REM")
IL 130  IF RP=0 THEN RP=INSTR(L$
       ,"'")
LC 140  IF RP THEN L$=LEFT$(L$,R
       P-1):IF LV=1 THEN L$=L$+"
       '"
AG 150  IF LV=2 AND RP=INSTR(L$,
       " ")+1 THEN 170
BC 160  PRINT#2,L$
EC 170 WEND
HP 180 CLOSE#1
IM 190 CLOSE#2
QP 200 PRINT:PRINT:PRINT"Finishe
       d."
LN 210 END
```

©

64 UN-CRUNCHER

Larry Dinwiddie

This convenient Commodore 64 utility "uncrunches" crowded BASIC program lines into separate, easily readable and edited lines. The utility works with either disk or tape, and although it is written in machine language, no machine language expertise is required to use it. It runs on any Commodore 64 or 128 (in 64 mode).

One common programming technique in Commodore BASIC is to "crunch" programs into compact form by combining multiple statements on a single program line. As most programmers soon learn, crunching conserves memory and helps a program run faster. In addition, programs listed in magazines and books are usually crunched to save space.

However, crunching also makes the program more difficult to read and modify. Often, modifying a crunched program involves breaking up a long line into two or more shorter lines. This can be tedious, and it increases the risk of errors.

"64 Uncruncher" automatically uncrunches an entire program for

you, making each BASIC statement a separate program line. The resulting program is much easier to modify than the original. And because each statement is a separate line, it is simpler to follow the program's logic as well.

Figures 1 and 2 illustrate a simple BASIC program before and after uncrunching. Both programs are listed with a width of 40 columns so they appear just as they would on your screen. Notice how much easier it is to read the uncrunched version and decipher its logic.

Figure 1: Crunched Program

```
10 POKE 53281,15:POKE 53280,15:POKE 646,
0:PRINT CHR$(147):FOR J=1 TO 10
20 GOSUB 50:IF INT(K/2)=K/2 THEN PRINT "
, AN EVEN NUMBER":GOTO 40
30 PRINT ", AN ODD NUMBER"
40 NEXT J:PRINT:PRINT "FINISHED":END
50 READ K:PRINT "K =" K;:RETURN:DATA 123
,456,789,987,654,321,123,456,789,111
```

Figure 2: Uncrunched Program

```
100 POKE 53281,15
110 POKE 53280,15
120 POKE 646,0
130 PRINT CHR$(147)
140 FOR J=1 TO 10
150 GOSUB 220
160 IF INT(K/2)=K/2 THEN PRINT ", AN EVE
N NUMBER":GOTO 180
170 PRINT ", AN ODD NUMBER"
180 NEXT J
190 PRINT
200 PRINT "FINISHED"
210 END
220 READ K
230 PRINT "K =" K;
240 RETURN
250 DATA 123,456,789,987,654,321,123,456
,789,111
```

Using Uncruncher

Since Uncruncher is written in machine language, you must type it in using the "MLX" machine language entry program listed elsewhere in this issue. Be sure you read and understand the instructions for using MLX before you begin entering the data for Uncruncher.

When you run the MLX program, it asks you for a starting address and an ending address. Here are the addresses for Uncruncher:

Starting address: C000
Ending address: C60F

After you've typed in and saved all of the Uncruncher data, you can test it on any Commodore 64 BASIC program. Follow these steps:

1. Load Uncruncher into memory by typing LOAD *"filename"*,8,1 for disk or LOAD *"filename"*,1,1 for tape. Substitute your own filename, of course.

2. Type NEW and press RETURN.

3. Load (but do not run) the BASIC program you want to uncrunch.

4. To start Uncruncher, type SYS 49152 and press RETURN. The screen clears and Uncruncher displays messages informing you of its progress. It takes three passes through the BASIC code to uncrunch the program. When the READY prompt returns, the uncrunching is complete.

You may pause Uncruncher at any time by holding down the f1 function key. Do not interrupt Uncruncher by pressing RUN/STOP–RESTORE; if you do, the program may be left in a garbled, unusable form. No real harm is done, but you'll need to reload the program and restart Uncruncher.

Uniform Line Numbers

Uncruncher begins numbering the new program at line 100 using line increments of 10 (110, 120, and so on). Each BASIC statement is a separate line, except for lines containing IF-THEN statements. Because the THEN portion of such a statement must be on the same line as IF, IF-THEN lines are left unchanged except for renumbering. For any BASIC statement that references a line number (such as GOTO, GOSUB, IF-THEN, RUN, and LIST), the line reference is also renumbered.

During the third pass, the program prints the line numbers it is replacing. If Uncruncher finds a statement that refers to a nonexistent line number, it prints this error message:

UNREFERENCED BRANCH IN NEW LINE # xxxxx

When you see this message, *xxxxx* is replaced by the new line number where the nonexistent reference is located. To mark where the error occurred, Uncruncher replaces the meaningless line number with 63999, the highest legal line number. An unreferenced branch error indicates a logic error in the original program, so you should reload the original, correct the error, and then repeat the uncrunching process.

If uncrunching generates a large number of unreferenced line number errors, you may find it useful to divert Uncruncher's output to a printer. To do this, make sure the printer is turned on, then enter this statement in direct mode (without a line number):

OPEN 4,4:CMD 4:SYS 49152

Now everything that would have been printed on the screen is sent to the printer instead. When Uncruncher is finished, type this statement and press RETURN:

PRINT#4:CLOSE 4

DATA statements are uncrunched in a special way. After uncrunching, each DATA line contains approximately 60 characters per line, including the line number and the keyword DATA. However, no individual DATA item will be split across two lines.

After the program has been uncrunched, you may list it, resave it, or modify it as usual. If you have a crunch utility, you may wish to recrunch the program after making modifications.

Because the additional line numbers take up more memory, the uncrunched program is significantly larger than the original, leaving less memory for BASIC variables and arrays. In most cases this should not cause a problem other than slowing program execution somewhat. However, a very large BASIC program or one that requires a great deal of variable space may not run correctly in uncrunched form.

Similar problems may arise if the original program POKEs sprite shapes, custom characters, or other data into a reserved area within BASIC memory. If the uncrunched program text expands into the reserved area, the POKEs may destroy part of the program text. To be on the safe side, you may want to save the uncrunched program immediately before you try to run it.

64 Uncruncher

Please refer to the "MLX" article in this issue before entering the following listing.

```
C008:8D 8D C5 A5 74 8D 8E C5 83
C010:A5 75 8D 8F C5 A9 4C 85 5F
C018:73 A9 15 85 74 A9 C5 85 14
C020:75 20 3A C5 A9 FF 91 A3 1C
C028:C8 91 A3 A9 9E A0 C5 20 A4
C030:1E AB A9 C0 A0 C5 20 1E 67
C038:AB A9 EF 8D 28 03 A9 00 71
```

```
C040:8D 90 C5 20 15 C5 AA 20 9C
C048:15 C5 E0 00 D0 07 C9 00 18
C050:D0 03 4C 09 C1 20 15 C5 93
C058:20 15 C5 EE 90 C5 20 15 C7
C060:C5 C9 00 F0 D9 C9 A7 F0 7C
C068:38 C9 CB F0 2A C9 8A F0 7F
C070:30 C9 9B F0 2C C9 A4 F0 C1
C078:23 A2 00 8E 91 C5 C9 8D E1
C080:F0 2C C9 89 F0 28 C9 22 35
C088:D0 D4 20 15 C5 C9 00 F0 43
C090:AD C9 22 D0 F5 F0 C7 EE 9E
C098:91 C5 D0 C2 AD 91 C5 F0 CA
C0A0:BD 20 15 C5 C9 30 90 B9 F1
C0A8:C9 3A B0 B5 90 03 20 15 F4
C0B0:C5 20 6B A9 85 02 A9 0B B7
C0B8:85 A3 A9 C6 85 A4 A0 00 87
C0C0:B1 A3 AA C8 B1 A3 E0 FF C3
C0C8:D0 04 C9 FF F0 15 E4 14 A6
C0D0:D0 04 C5 15 F0 21 18 A5 A7
C0D8:A3 69 04 85 A3 90 DF E6 65
C0E0:A4 D0 DB 88 A5 14 91 A3 31
C0E8:A5 15 C8 91 A3 A9 FF A2 1B
C0F0:04 C8 91 A3 CA D0 FA A5 48
C0F8:02 F0 0B C9 2C F0 AF C9 04
C100:AB F0 AB 4C 61 C0 4C 3E B4
C108:C0 A9 C9 A0 C5 20 1E AB 30
C110:20 3A C5 A2 00 8E 90 C5 36
C118:A9 0B 85 A3 A9 C6 85 A4 36
C120:20 15 C5 AA 20 15 C5 E0 1E
C128:00 D0 07 C9 00 D0 03 4C F2
C130:FB C1 20 15 C5 85 FB 20 D3
C138:15 C5 85 FC A2 01 8E 90 FE
C140:C5 A0 00 B1 A3 AA C8 B1 F4
C148:A3 E0 FF D0 04 C9 FF F0 1B
C150:1F E4 FB D0 04 C5 FC F0 4B
C158:0D 18 A5 A3 69 04 85 A3 61
C160:90 DF E6 A4 D0 DB C8 A5 78
C168:A5 91 A3 C8 A5 A6 91 A3 B2
C170:20 15 C5 C9 00 F0 66 C9 F8
C178:3A F0 55 C9 22 F0 0C C9 53
C180:83 F0 15 C9 88 D0 E9 E6 9B
C188:A7 D0 E5 20 15 C5 C9 00 26
C190:F0 4B C9 22 D0 F5 F0 D8 D3
C198:A0 00 8C 90 C5 20 15 C5 A5
C1A0:C8 C9 22 F0 1D C9 00 F0 4F
C1A8:34 C9 3A F0 23 C9 2C D0 78
C1B0:EC C0 32 90 E8 18 A5 A5 C2
C1B8:69 0A 85 A5 90 DA E6 A6 E2
C1C0:D0 D6 20 15 C5 C8 C9 00 9C
C1C8:F0 13 C9 22 D0 F4 F0 CD EE
C1D0:A2 01 8E 90 C5 85 02 A5 AE
C1D8:A7 D0 95 F0 09 A2 01 8E 89
C1E0:90 C5 85 02 85 A7 18 A5 8F
C1E8:A5 69 0A 85 A5 90 02 E6 8D
C1F0:A6 A5 02 D0 03 4C 13 C1 AF
C1F8:4C 70 C1 A9 D1 A0 C5 20 4E
C200:1E AB 20 3A C5 A9 00 85 81
C208:A7 8D 90 C5 20 15 C5 AA BE
C210:20 15 C5 E0 00 D0 07 C9 CC
C218:00 D0 03 4C 67 C5 AD 24 C8
C220:C5 85 FB AD 25 C5 85 FC 8C
C228:A0 01 A5 A5 91 FB C8 A5 01
C230:A6 91 FB 18 A5 A5 69 0A 0F
C238:85 A5 90 02 E6 A6 20 15 43
C240:C5 8D 94 C5 20 15 C5 8D 69
C248:95 C5 EE 90 C5 20 15 C5 8F
C250:C9 00 F0 B1 C9 3A F0 4D 5A
C258:C9 22 F0 39 C9 CB F0 29 85
C260:C9 A4 F0 28 A2 00 8E 91 58
C268:C5 C9 83 F0 49 C9 8A F0 3A
C270:1E C9 9B F0 1A C9 A7 F0 32
C278:16 C9 8D F0 15 C9 89 F0 10
C280:11 C9 8B D0 C8 E6 A7 D0 81
C288:C4 4C ED C2 4C F3 C2 4C 71
C290:FB C2 4C 09 C3 20 15 C5 6D
C298:C9 00 F0 06 C9 22 D0 F5 EF
C2A0:F0 AB 4C 05 C2 A2 01 8E 94
C2A8:90 C5 A5 A7 D0 9F A9 04 73
C2B0:20 F9 C4 4C 05 C2 A0 00 96
C2B8:8C 90 C5 20 15 C5 C8 C9 7E
C2C0:00 F0 DF C9 3A F0 DE C9 38
C2C8:22 F0 14 C9 2C D0 EC C0 F9
C2D0:32 90 E8 A9 05 20 F9 C4 AC
C2D8:A9 83 91 A8 4C 05 C2 20 EC
C2E0:15 C5 C8 C9 00 F0 B8 C9 1D
C2E8:22 D0 F4 F0 CE EE 91 C5 7C
C2F0:4C 4D C2 AE 91 C5 D0 03 7B
C2F8:4C 4D C2 20 15 C5 C9 30 D5
C300:90 04 C9 3A 90 06 4C 50 33
C308:C2 20 15 C5 AE 24 C5 86 10
C310:A8 AE 25 C5 86 A9 20 6B 1F
C318:A9 85 02 A9 0B 85 A3 A9 10
C320:C6 85 A4 A0 00 81 A3 AA C3
C328:C8 B1 A3 E4 14 D0 04 C5 F4
C330:15 F0 0D 18 A5 A3 69 04 34
C338:85 A3 90 E7 E6 A4 D0 E3 4B
C340:C8 81 A3 AA 85 63 C8 B1 84
C348:A3 85 62 E0 FF D0 21 C9 AC
C350:FF D0 1D A9 DA A0 C5 20 4F
C358:1E AB AE 94 C5 AD 95 C5 CE
C360:20 CD BD A9 0D 20 D2 FF 4C
C368:A9 F9 85 62 A9 FF 85 63 D5
C370:A2 90 38 20 49 BC 20 DF D3
C378:8D 85 F8 84 FC A0 00 B1 24
C380:FB 99 98 C5 F0 03 C8 D0 D1
C388:F6 A9 FE A0 C5 20 1E AB 76
C390:EE 20 D0 A9 98 A0 C5 20 3F
C398:1E AB 38 AD 24 C5 E5 A8 A8
C3A0:85 FD A0 00 89 98 C5 C9 04
C3A8:00 F0 03 C8 D0 F6 8C 97 6C
C3B0:C5 C4 FD F0 25 B0 0D A5 C6
C3B8:FD 84 FD 38 E5 FD 20 F8 04
C3C0:C3 4C DA C3 38 98 E5 FD C2
C3C8:20 6C C4 18 AD 24 C5 6D 8C
C3D0:96 C5 8D 24 C5 90 03 EE 6E
C3D8:25 C5 A0 00 AE 97 C5 89 91
C3E0:98 C5 91 A8 C8 CA D0 F7 ED
C3E8:A5 02 C9 2C F0 07 C9 AB A2
C3F0:F0 03 4C 50 C2 4C 09 C3 5D
C3F8:A6 A8 86 FD A6 A9 86 FE 96
C400:8D 96 C5 18 65 FD 85 FD 5C
C408:90 02 E6 FE A5 2D 85 AA BE
C410:A5 2E 85 AB 38 A5 AA E5 F6
C418:FD 85 AC B0 02 C6 AB 38 5D
C420:A5 AB E5 FE AA A0 00 B1 9D
C428:FD 91 A8 C8 C4 AC D0 F7 29
C430:A0 00 E0 00 F0 0E E6 FE B2
C438:E6 A9 81 FD 91 A8 C8 D0 47
C440:F9 CA D0 F2 38 A5 2D ED 63
C448:96 C5 85 2D B0 02 C6 2E 5B
C450:38 A5 2F ED 96 C5 85 2F 2A
C458:B0 02 C6 30 38 AD 24 C5 1D
C460:ED 96 C5 8D 24 C5 B0 03 B4
C468:CE 25 C5 60 8D 96 C5 A6 5A
C470:2D 86 A3 86 AA A6 2E 86 E1
C478:A4 86 AB A6 A8 8E 92 C5 40
C480:A6 A9 8E 93 C5 38 A5 AA D7
C488:E5 A8 85 FD B0 02 E6 A9 C4
C490:38 A5 AB E5 A9 85 FE 38 0D
C498:A5 A3 E5 FD 85 A3 85 AA EA
C4A0:80 04 C6 A4 C6 AB AD 96 7D
C4AB:C5 A4 FD 18 65 A3 85 A3 E7
C4B0:90 02 E6 A4 81 AA 91 A3 29
C4B8:88 D0 F9 81 AA 91 A3 A6 9E
C4C0:FE F0 13 C6 AB C6 A4 B8 1F
C4C8:B1 AA 91 A3 88 D0 F9 B1 6F
C4D0:AA 91 A3 CA D0 ED AD 96 65
C4D8:C5 18 65 2D 85 2D 90 02 CE
C4E0:E6 2E AD 96 C5 18 65 2F 11
C4E8:85 2F 90 02 E6 30 AE 92 1B
C4F0:C5 86 A8 AE 93 C5 86 A9 69
C4F8:60 AE 24 C5 86 A8 AE 25 98
C500:C5 86 A9 20 6C C4 A9 00 11
C508:A8 AE 96 C5 91 A8 C8 A9 2D
C510:FF CA D0 F8 60 A5 CB C9 F2
C518:40 D0 FA EE 24 C5 D0 03 23
C520:EE 25 C5 AD 01 08 AE 90 16
C528:C5 F0 0E C9 3A B0 0A C9 A3
C530:20 F0 E2 38 E9 30 38 E9 52
C538:D0 60 A9 0B 85 A3 A9 C6 FE
C540:85 A4 A9 64 85 A5 A9 00 49
C548:85 A6 85 A7 A5 2B 8D 24 84
C550:C5 A5 2C 8D 25 C5 38 AD E4
C558:24 C5 E9 01 8D 24 C5 B0 ED
C560:03 CE 25 C5 A0 00 60 A5 8D
C568:2F 85 31 A5 30 85 32 A9 13
C570:ED 8D 28 03 20 33 A5 AD 52
C578:8D C5 85 73 AD 8E C5 85 DC
C580:74 AD 8F C5 85 75 AD 0A 67
C588:C6 8D 20 D0 60 00 00 00 EE
C590:00 00 00 00 00 00 00 00 1C
C598:20 20 20 20 20 20 93 11 FB
C5A0:0E 2A 2A 2A 20 C3 CF CD 23
C5A8:D0 D5 D4 C5 21 27 53 20 75
C5B0:D5 4E 43 52 55 4E 43 48 FA
C5B8:45 52 20 2A 2A 2A 0D 00 36
C5C0:11 D0 41 53 53 20 31 0D F0
C5C8:00 D0 41 53 53 20 32 0D 72
C5D0:00 D0 41 53 53 20 33 0D 7C
C5D8:11 00 0D D5 4E 52 45 46 78
C5E0:45 52 45 4E 43 45 44 20 09
C5E8:42 52 41 4E 43 48 20 49 FB
C5F0:4E 20 4E 45 57 20 4C 49 E6
C5F8:4E 45 20 23 20 00 0D 91 DF
C600:20 20 20 20 20 20 20 0D 7A
C608:91 00 00 00 00 00 00 00 5E
```

©

COMPUTE!
TOLL FREE
Subscription
Order Line
1-800-247-5470
In IA 1-800-532-1272

This Publication is available in Microform.

University Microfilms International

Please send additional information
for _____
Name _____
Institution _____
Street _____
City _____
State _____ Zip _____

300 North Zeeb Road
Dept P.R.
Ann Arbor, Mi 48106

Do-It-Yourself Movies On An Apple

Recently my ten-year-old daughter Catie asked if I'd like to help her with her school science project. Oh, boy! I thought. Here's a chance to show her how she could take advantage of a computer!

I was almost afraid to suggest that we use a computer, however. She's not quite as fanatical about the machines as I am, and she gets tired hearing how every family activity can somehow be tied to computers. So I didn't mention the word "computer" at all. Instead, I said, "Hey, Catie, how'd you like to make a movie for your science project?" This idea delighted her, so off we went.

The first step was to choose a subject. Catie chose black holes. "Okay," I said. "You have to do two things: Draw a bunch of squares like you see in the funnies in the newspaper, and draw pictures inside the squares of the black hole—how it's born, how it grows, and so on. Next, sit down and write a script for the movie. Match what you say in the script with the pictures in the squares."

Frame By Frame

Catie raced off and drew the pictures and wrote her script. When she came back, I was sitting in front of our Apple IIc. "Daddy," she said, "why are you sitting at the computer? We're supposed to be working on my movie."

"Aha!" I said. "The computer is going to help us make that movie." I introduced her to a program called *Fantavision*. *Fantavision* looks like a normal drawing program—it has a drawing window surrounded by lots of tools and menu options around the border. I showed Catie how she could draw things freehand or with rubberband lines, squares, circles, and so on. She could fill the objects with color, stretch them, rotate them, squish them, cut them, and paste them anywhere in the window.

But this was just the beginning. When she was done creating a picture of a happy face, I showed her how she had just created one frame in a cartoon. She could use the mouse to scroll the screen and begin creating the next frame. Catie then drew a face of a kitty cat.

"And now you've got a little movie," I told her. I pointed the mouse to a menu box labeled GO, and we watched a short cartoon of the happy face changing into the face of the cat.

A whole movie from only two frames? The secret is a complex technique that animators call *tweening* (derived from *between*). *Fantavision* automatically constructed dozens of new frames from Catie's first frame and second frame, then inserted them between her frames to smooth out the transition. These new frames, called tweeners, made the happy face in her first frame change gradually into the kitty's face of her second frame.

From Giant To Dwarf

With very little help from me, Catie sat down at the computer and learned how to use *Fantavision* in about half an hour. She copied her hand-drawn frames from her notebook onto the screen. The first frame was a picture of a normal, yellow-looking sun surrounded by stars in outer space. The next frame was of the same sun, now billions of years older, swollen to become a red giant star. The third frame showed the star shrunken into a tiny white dwarf star.

The white dwarf continued to shrink until it became a black hole. Catie drew a picture of the black hole that was straight out of Walt Disney—with swirling white clouds of cosmic gas spiraling around a dark center. Next, using the COPY, MOVE, and ROTATE commands, she drew successive frames of the black hole rotating and gobbling up stars.

Then Catie designed a title, which turned out to be one of the most spectacular parts of the movie. By using the ZOOM command, Catie was able to create several successive frames with the words "The Black Hole" growing larger and larger. And when the movie starts, the letters in the title break up into pieces which come together to form the stars and the sun. This looks like an amazing special effect, but it was completely unintended; it was just a by-product of *Fantavision*'s tweening capabilities.

Finally, Catie and I set up the Apple in the room with the stereo cassette player. We bought a copy of the soundtrack to the movie *Jaws* and aimed a video camera at the computer screen while the hungry shark music was playing in the background. Catie read her script as the movie progressed—from the opening title to the birth and growth of the black hole. It took us several tries to synchronize the music, Catie's narration, and the *Fantavision* movie, but it was well worth it. The next day, Catie took her project to school and won a blue ribbon for her efforts.

I was really proud of her, but my biggest thrill came when she ran up to me after the judging and said, "You know, Daddy, sometimes I'm glad that our family has a computer."

(*Fantavision* is available for $49.95 for all Apple II-series computers with at least 64K of RAM. For more information, contact Brøderbund Software, 17 Paul Drive, San Rafael, CA 94903.) ©

Computers and Society

David D Thornburg, Associate Editor

Speak Softly And Carry A Big RAM

Sooner or later it was inevitable that the grandiose and bizarre claims of some computer scientists would result in a critical response from someone with a radically different point of view. I just read such a response by Theodore Roszak, a history professor at California State University, Hayward. Roszak's book, *The Cult of Information* (Pantheon Books, 1986), lambastes an entire field for the excesses of a few. As a result, he is guilty of the same error as the people he criticizes—members of the artificial intelligensia who claim that computers accurately mimic human behavior and who feel we would be better off learning to think like machines.

I share his distaste for the extravagant and largely unsupported claims made by those who feel that silicon consciousness is our evolutionary destiny. What distresses me is that Roszak expresses the belief that the general acceptance of computers into our homes, schools, and workplaces is somehow damaging to our identity as human beings.

He is confusing the tool with the result and forgetting that technology is inherently neutral. Computers can be (and have been) used in inappropriate ways, as have fountain pens. How easy it is to use the wild exhortations of our field's fringe fanatics to damn an entire technology—one that most of us understand and use without feeling any loss of humanity.

Towards Holistic Thought

Rather than diminishing our human qualities, I think computers allow us to integrate our thinking—to become holistic learners who see knowledge as something more than a collection of facts stored in separately labeled boxes, each with its own content, and with little or no connection between them. The specialization of knowledge into fields was a result of an information explosion that made it impossible for one person to achieve mastery of all subjects. While this division will remain important for many experts in these fields, there is increasing evidence that it has negative consequences.

To take one example, many particle physicists are finding that advances in their field are aided by a study of Taoist philosophy. My own hobby of computational linguistics is populated by linguists, philosophers, and computer scientists, each willing to learn from the others. True knowledge is interdisciplinary. As soon as one draws a box around a topic to clarify the object of study, one risks excluding information or viewpoints that can end up being quite important.

I sometimes put on a multimedia event called *The Magic Universe of Recursion* in which I show the appearance of this mathematical concept in computer programming, music, art, literature, philosophy, and religion. My point is not to say that each of these topics is mathematical—mostly they are not. Rather, it is my purpose to show that recursion is an idea that breaks across traditional barriers of knowledge.

I am convinced that there are hundreds of general concepts that transcend the fields in which they were first used, if only we would look for them. Fortunately, one tool to aid in our search is readily at hand—the personal computer.

Enter The Computer

As a tool that lets us manipulate information and construct metaphorical worlds of our own design, the computer can help us chart a path across the boundaries of numerous disciplines in our quest for holistic education.

Most fields of endeavor are so complex and demanding that one has little time to search beyond the walls of one subject for ideas of value from another area. As computer technology becomes easier to use, the tedious aspects of at least some parts of many fields will be relegated to machinery, thus freeing people to stand back and take in a larger view of the subject. It is hard to take a reading on the stars while you are rowing the boat.

This is why I first became interested in Logo. I saw, in Logo's turtle graphics, a tool that would let me explore the mathematics of naturally occurring patterns. I have spent years exploring everything from cracks in drying mud to the delicate patterns in ferns. The ease with which I could generate, test, and evaluate hypotheses with the aid of the computer allowed me to ask questions I would not have dared to ask otherwise.

My point is that the really exciting uses of computers are likely to come from the interdisciplinary holistic thinkers—people who sense the unity behind the major ideas of our time and place. These people tend not to be technologists, because the intense study needed to master technology leaves (we are told) little room for anything else. The people I have in mind are those whose interests span many fields—physics and poetry, art and archaeology—people who probably have degrees in the "liberal arts."

In order for such people to use computer technology effectively, computers must have speed, lots of memory, excellent software, and a transparent user interface. Computers like the Macintosh and Amiga are stepping stones in the right direction. Software for these computers is being designed to rise to the level of the way people work rather than dragging the user down to the machine's level. ©

The Beginners Page

Tom R. Halfhill, Editor

Advanced String Features

To wrap up our long-running series on character strings in BASIC, let's take a look at some advanced string features which are finding their way into the latest and most sophisticated versions of the BASIC language. Although these features may not be found in the BASIC you're working with now, you'll probably encounter them sometime in the future.

If you want to keep up with these trends, pay attention to any new version of BASIC released by Microsoft, Inc. Microsoft certainly isn't the only company in the BASIC business, but it definitely *is* the market leader. Versions of Microsoft BASIC are either standard equipment or available as an option for almost every microcomputer ever made. When an advanced feature is introduced in a new version of Microsoft BASIC, it tends to cross over into the next version which is released, even if the next version is for a completely different computer.

For instance, the latest Microsoft BASIC to appear is Amiga BASIC. It shares numerous features with its nearest predecessor, Microsoft BASIC for the Macintosh. These two dialects are so much alike that some example programs in the Macintosh BASIC manual—even those with graphics—will run unchanged on the Amiga.

Super Strings

One new trend in Microsoft BASIC is to remove the 255-character limit on strings. Macintosh BASIC and Amiga BASIC both let you define strings up to 32,767 characters long.

So what? you might say. Who needs to display a message that's thousands of characters long? It probably won't fit on the screen, anyway.

But strings are good for lots of things besides displaying messages, of course, especially if they aren't limited to 255 characters. Programmers on the Atari 400/800/XL/XE computers know this well, because Atari BASIC has allowed super strings since 1979.

For instance, suppose you want to write a simple terminal program for downloading public domain software from information services and bulletin boards. Unless you're handy with a memory map, you might have trouble finding a large area of free memory in your computer to temporarily hold the downloaded data before storing it on disk. With super strings, it's no problem. Simply download everything into a single string, perhaps called BUFFER$. Since BASIC reserves and protects memory for the string, you don't have to worry about memory conflicts.

Best of all, the new Microsoft BASICs don't force you to give up anything in return for super strings—unlike Atari BASIC, which doesn't allow string arrays as a tradeoff for this feature.

Search And Replace

Another powerful feature of late-model BASICs is the INSTR function (pronounced *in-string*). INSTR searches through a longer string in search of any shorter string you specify. If INSTR finds the shorter string (*substring*), it returns a number indicating the substring's starting character position within the longer string. Example:

```
10 MAIN$="This is the longer string."
20 X=INSTR(MAIN$,"the")
```

When you run this program, INSTR returns the value 9 in the variable X, because the substring *the* begins at the ninth character position within MAIN$. Of course, you could also use a string variable for the substring parameter in the INSTR function. If INSTR can't find the substring, it returns a 0.

By adding another parameter, INSTR can be made to begin its search at any point within MAIN$. For example, X=INSTR(5,MAIN$,SUB$) would begin searching for SUB$ at the fifth character position of MAIN$. The INSTR function makes it a snap to write filing programs with rapid search-and-retrieve features, because it works at nearly machine language speed.

Some recent BASICs (including Macintosh BASIC, Amiga BASIC, and BASIC 7.0 on the Commodore 128) allow the use of MID$ as a statement as well as a function. You'll recall from the April column that the MID$ function lets you copy a substring from within a larger string. When used as a statement, MID$ lets you *replace* a specified substring with another string. And the replacement string isn't limited to the length of the substring it's replacing. When coupled with INSTR, the MID$ statement makes it easy to add a search-and-replace feature to a filing program.

Finally, another useful string command found in newer BASICs is UCASE$. This converts any string of lowercase characters to uppercase. Example:

```
PRINT UCASE$("capitalized")
```

results in CAPITALIZED. A logical application for the UCASE$ command is to make an INSTR search routine insensitive to case. For instance, the statement X=INSTR(UCASE$(MAIN$),UCASE$(SUB$)) will make certain that INSTR will find any matching SUB$ within MAIN$, even if some of the characters are mixed uppercase and lowercase.

Watch for more features like these to keep appearing in new versions of BASIC. Although it's over 20 years old, BASIC is only now experiencing its greatest growth spurt as programmers continue demanding more and more power from this popular language. ©

Arlan R. Levitan

This Fido's No Dog

In June 1984, Tom Jennings of San Francisco and John Madill of Baltimore began developing and testing an MS-DOS-based electronic bulletin board system (BBS) called Fido. Although Fido sported the usual file upload and download facilities, its electronic mail system was far from typical. Fido systems were not designed to exist as separate, isolated entities like most BBSs. Instead, Jennings and Madill set out to create a BBS that could network with others of its own kind. Rather than requiring users and system operators to call each other's BBSs to leave messages, Fido would routinely store and forward messages to other Fidos via modem in the dead of the night, when long-distance phone rates are lowest.

By August 1984 there were almost 30 Fido systems (commonly referred to as *nodes* by Fido fans). Since then, Fido has grown faster and bigger than a Saint Bernard. Today, more than 100,000 users communicate over FidoNet, which consists of more than 1,000 Fido systems spread across the U.S., Europe, and Australia. Using FidoNet, these telecomputing enthusiasts can communicate with each other overnight. And in addition to the public FidoNet, internal Fido systems are being widely used by private industry and government bureaus.

The sheer magnitude of FidoNet easily qualifies it as the largest publicly owned and operated telecomputing network in the world. Other attempts at nationwide networking via BBS have collapsed under their own administrative weight. But the organizational talents of Fido's creators and a dedicated inner core of Fido system coordinators and directors have been put to good use. Careful planning and more sweat than expended in a dozen NBA playoffs have kept the Fido network functioning smoothly.

Global Party Line

If you live in a metropolitan area, your local Fido is likely to be a member of a group of Fido systems located relatively close to each other. Each group is considered to be a *local network.* One system within the group is designated as a *network host.* The system operator of the network host is charged with maintaining a list of the nodes in the local net.

How does Fido work? During the day, Fido users can leave messages for both local and remote users. At about 4:30 a.m., the nodes within the local network begin dialing their network host to transfer messages intended for remote Fido systems. Once all of the outgoing messages from the local net have been collected, the network host compresses them to shorten transmission time, then starts calling other network hosts to send the messages. From 5:00 to 5:30 a.m., the network hosts dial up their local nodes to deliver incoming messages. Heavily used local nets often have two network hosts, one each for outgoing and incoming traffic.

Fidos that are too isolated to be a member of a local network are called *independents* and are permitted to forward and receive mail directly to and from network hosts and other independents. Regional Fido coordinators are responsible for keeping track of independents and encouraging them to join existing nets or forming new ones.

At this writing, the U.S. is divided into 12 Fido regions. Europe has six regions; Australia, two. There are 82 network hosts worldwide. Each host has an average local net of about 13 systems. It's interesting to note that the network hosts in Europe are equipped with two different types of modems. To handle local traffic, they use modems adhering to the CCITT (Consultative Committee on International Telephony & Telegraphy) standard, which employs different frequencies than our domestic units; U.S.-type modems handle transfers to and from Fidos in North America.

Managing The FidoNet

What's truly amazing is that the cost of operating FidoNet is very low when spread out over the entire user base. There is no centralized billing. The local nodes are at liberty to recoup the long-distance charges incurred by their network host however they see fit, either footing the bill themselves or by charging a small yearly membership fee to their local users.

The logistics of keeping things straight within FidoNet could turn into a never-ending "Who's on first?" dilemma if everyone didn't have a constantly updated "phone book," or node list, for all of the systems. Network host operators and regional coordinators are responsible for notifying the national Fido coordinators of any changes in their networks. The national coordinators, in turn, forward a compiled list of changes to Ken Kaplan, executive director of the International Fido Association. A list of FidoNet changes is automatically transmitted to the network hosts every weekend from Kaplan's Fido.

There's also an excellent weekly FidoNet newsletter, managed by FidoFiend Tom Henderson, that's both compiled and distributed via FidoNet. For more information, write to the International FidoNet Association, P.O. Box 41143, St. Louis, MO 63141. ©

INSIGHT: Atari

Bill Wilkinson

Many people have asked me to discuss the use of the DOS 2.5 RAM disk with Atari computers other than the 130XE. Most are interested either in one of the 800XL memory upgrade kits now available or in simply using the extra 16K memory of an XL as a very small RAM disk. Since I've seen the subject treated incorrectly in several user group newsletters, I decided that some mildly technical discussion here would not be amiss.

Many months ago, in one of my columns, I described the memory map of an Atari XL computer. This time, let's see how a 130XE is a fairly simple expansion of the XL models.

An Atari 130XE has 126K—*not* 128K—of Random Access Memory. (Keep in mind that one kilobyte equals 1,024 bytes.) The first 62K is used and accessed exactly the same way as the 62K in the 1200XL and 800XL (that *62K* is not a typo, either—more on this later). Now, a 6502 microprocessor can address a total of only 64K of contiguous memory, because the address counter goes from 0 to 65535 (64 * 1024). In the hexadecimal (base 16) numbering system used by computers, those addresses are expressed as $0000 (0) to $FFFF (65535). When the address counter passes $FFFF, it rolls back to $0000 again. This is kind of like a car speedometer which only goes to 99,999.9 miles; another tenth of a mile and you have a new car again.

So, how does the 130XE access its extra 64K of memory? By a technique known as *bank selection.*

Cashing In At The Memory Bank

The extra 64K in the 130XE is divided into four separate 16K banks. The 6502 can access only one of these banks at a time. But wait, you say, if the main memory uses up the full addressable range of the 6502, where do these extra banks fit in?

The answer: 16K of the main memory (that is, of the regular 64K) is disabled. Effectively, then, a 130XE has *five* banks of RAM, each consisting of 16K, plus another 46K (not a typo) which is *not* bank selected.

Now comes the important part: Just where, within the 64K address space of the 6502, are these five banks addressed? As Appendix H of the 130XE owner's manual states, the selectable 16K bank falls between locations 16384 (hex $4000) and 32767 ($7FFF). This is the second quarter of the 6502's 64K memory space. Why was this 16K area chosen instead of some other area? Because the first quarter of memory includes zero page, and bank-selecting zero page is a tricky proposition in a computer which is handling interrupts. The other two quarters of memory share space with cartridges and/or the operating system ROM, which would make programming more complicated. Thus, the second quarter of memory wins by default.

Okay so far. Now let's consider where BASIC programs reside in memory. Generally they begin at a memory location called LOMEM, which can vary but is usually between $1C00 and $2400 (about 7000 and 9000) when DOS is booted. BASIC programs always end below screen memory, which in turn is below the BASIC ROM. In practice, this means that BASIC programs and their variables are limited to a length of about 31K.

Let's assume that LOMEM is at $2000 (8192). Let us also assume that we have loaded or typed in a BASIC program which is 12,000 bytes long. Where does that program end? Smack dab in the middle of the second quarter of memory, where the banks are selected.

You might think that this would cause a problem on a 130XE, since it has to switch that bank of memory on and off. But it's not a problem, because one of those five banks is assigned to be main memory—that is, the memory corresponding to the only memory at that address in a 1200XL or 800XL. The DOS 2.5 RAM disk never touches that bank; it limits itself to the other four banks.

Okay, enough background on the 130XE. Is there a way to use the extra 16K memory of the 800XL as a RAM disk? Yes, but it isn't easy. That extra memory is addressed from $C000 to $FFFF (but see below for an exception). Aside from the fact that DOS 2.5 wasn't designed to see a RAM disk in this address range, this range is shared with the operating system ROMs and the hardware input/output area. Shared? Yep, more bank selection. And this bank is even trickier to use.

To Be Continued

Just as things start to get interesting, I run out of room. There is much more to this topic. For example, we haven't even looked for the missing 2K of RAM in the XLs and XEs, have we? And wouldn't it be nice to consider the effects of some of the add-on memory kits for the XLs? Until next month, let me tantalize you with some tidbits.

The RAM disk which emulates drive 8 (D8:) is one of the nice features of DOS 2.5. One of the not-so-nice features is that the RAM disk is *always* D8:. Many, many programs which want two disk drives assume that the second drive is D2:. Wouldn't it be nice if we could change the RAM disk's drive number? Say no more. The BASIC program listings below accomplish this for you.

Program 1, "REPLACE.BAS,"

is for use with the RAMDISK.COM program supplied with DOS 2.5. After you boot the system with DOS 2.5 and RAMDISK.COM, this program simply changes all the magic memory locations in DOS so that the RAM disk is now addressed as D2:. (Or you can change lines 190 and 260 to make the RAM disk emulate any drive from D2: to D8:.) If you use Program 1, the DOS files DUP.SYS and MEM.SAV will be on D2:, but otherwise DOS 2.5 will be unchanged.

Program 2, "MAKERAM.BAS," serves another purpose. As you've probably noticed, DUP.SYS and MEM.SAV take up a lot of room on the RAM disk. True, keeping them on the RAM disk does make DOS easier to use. However, if your program won't use DOS but could use more RAM disk space, why not leave them on D1:? That's exactly what MAKERAM.BAS does. It initializes and installs the RAM disk, but copies no files to it—all 499 RAM disk sectors are available for your use. Naturally, you may choose any drive number for the RAM disk (see lines 190 and 260 again). And, although we could change this program to allow it to work *after* RAMDISK.COM has booted, it is a waste of time since this program reinitializes the RAM disk, anyway. Therefore, you should erase or rename the RAMDISK.COM file when using MAKERAM.BAS (but don't erase your only copy of RAMDISK.COM).

Finally, Program 3, "MAKERAM.SUB," simply changes Program 2 into a subroutine which you can include in your own programs. Use it anytime you want your program to initialize a blank RAM disk.

For instructions on entering these listings, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!

Program 1: REPLACE.BAS

```
HF 100 REM
DA 110 REM ==== REPLACE.BAS
       ====
HH 120 REM
OC 130 REM A program to repl
       ace D8: with
CM 140 REM Dn: where n is an
       y drive
BL 150 REM number from 2 to
       7 (or even 8)
HL 160 REM
AI 170 IF PEEK(1802)<128 THE
       N PRINT "No RamDisk i
       nstalled!":STOP
HN 180 REM
CD 190 RAMDRIVENUM=2:REM Cha
       nge this as desired
HG 200 REM
AE 210 POKE 1920,RAMDRIVENUM
AN 220 POKE 2953,RAMDRIVENUM
KG 230 POKE 5439,48+RAMDRIVE
       NUM
MF 240 POKE 1802,PEEK(1802)-
       128
LK 250 REM (for changes to l
       ine 260, see "Mapping
        the Atari")
OK 260 IF PEEK(1802)=1 AND R
       AMDRIVENUM=2 THEN POK
       E 1802,3
FA 270 DIM INIT$(4)
HP 280 FOR I=1 TO 4:READ DAT
       A
PI 290 INIT$(I)=CHR$(DATA):N
       EXT I
AC 300 DATA 104,76,224,7
NE 310 JUNK=USR(ADR(INIT$))
HJ 320 REM
FI 330 REM Verify it worked
HL 340 REM
JH 350 DIM DRIVE$(6)
CG 360 DRIVE$="Dn:*.*"
IJ 370 DRIVE$(2,2)=CHR$(48+R
       AMDRIVENUM)
HP 380 REM
KK 390 OPEN #1,6,0,DRIVE$
GC 400 TRAP 430
LG 410 GET #1,BYTE:PRINT CHR
       $(BYTE);
GE 420 GOTO 410
GD 430 END
```

Program 2: MAKERAM.BAS

```
HF 100 REM
GA 110 REM ==== MAKERAM.BAS
       ====
HH 120 REM
IO 130 REM A program to set
       up a RAM disk on
FI 140 REM Dn:, where n is a
       ny drive
BL 150 REM number from 2 to
       7 (or even 8)
HL 160 REM
CD 170 IF PEEK(1802)>127 THE
       N PRINT "RamDisk alre
       ady installed!":STOP
HN 180 REM
CD 190 RAMDRIVENUM=2:REM Cha
       nge this as desired
HG 200 REM
AE 210 POKE 1920,RAMDRIVENUM
AN 220 POKE 2953,RAMDRIVENUM
OC 230 POKE 5439,49
MO 240 REM (line 230 forces
       DUP.SYS to drive 1)
LK 250 REM (for changes to l
       ine 260, see "Mapping
        the Atari")
OK 260 IF PEEK(1802)=1 AND R
       AMDRIVENUM=2 THEN POK
       E 1802,3
FA 270 DIM INIT$(4)
HP 280 FOR I=1 TO 4:READ DAT
       A
FI 290 INIT$(I)=CHR$(DATA):N
       EXT I
AC 300 DATA 104,76,224,7
NE 310 JUNK=USR(ADR(INIT$))
HJ 320 REM
JF 330 DIM DRIVE$(6)
CE 340 DRIVE$="Dn:*.*"
IH 350 DRIVE$(2,2)=CHR$(48+R
       AMDRIVENUM)
HN 360 REM
NL 370 REM Initialize our ne
       w disk
HP 380 REM
CJ 390 XIO 254,#1,0,0,DRIVE$
HI 400 REM
FH 410 REM Verify it worked
HK 420 REM
KF 430 OPEN #1,6,0,DRIVE$
GK 440 TRAP 470
LF 450 GET #1,BYTE:PRINT CHR
       $(BYTE);
GM 460 GOTO 450
HC 470 END
```

Program 3: MAKERAM.SUB

```
BD 10 GOSUB 9000:REM Your pr
      ogram here
DJ 20 END
KN 9000 REM
KM 9010 REM ==== MAKERAM.SUB
        ====
KP 9020 REM
HP 9030 REM Subroutine to se
        t up RAM disk
LG 9040 REM
EE 9050 IF PEEK(1802)>127 TH
        EN PRINT "Disk alrea
        dy installed!":STOP
CI 9060 POKE 1920,2
DA 9070 POKE 2953,2
GO 9080 POKE 5439,49
CL 9090 POKE 1802,3
FI 9100 DIM RAMDISK$(4)
PO 9110 FOR N=1 TO 4:READ X
EI 9120 RAMDISK$(N)=CHR$(X):
        NEXT N
DM 9130 DATA 104,76,224,7
OF 9140 JUNK=USR(ADR(RAMDISK
        $))
PM 9150 REM (any handy strin
        g can be used instea
        d of DRIVE$)
MP 9160 DIM DRIVE$(6)
CC 9170 DRIVE$="D2:*.*"
FP 9180 XIO 254,#1,0,0,DRIVE
        $
LO 9190 RETURN                  ©
```

Attention Programmers

COMPUTE! magazine is currently looking for quality articles on Commodore, Atari, Apple, and IBM computers (including the Commodore Amiga and Atari ST). If you have an interesting home application, educational program, programming utility, or game, submit it to COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. Or write for a copy of our "Writer's Guidelines."

Donald B. Trivette

WW II And KQ III

GATO is one of the most interesting games to come along for the IBM PC, PCjr, and compatibles in the last year. It's a strategy game that puts you in the captain's seat of a World War II *Gato*-class submarine. Your mission may be to rescue a downed pilot, resupply a friendly coast watcher, or sink an enemy fleet. Once you receive your orders, you must pilot your boat through enemy waters and around dangerous reefs using radar, charts, and the periscope—if you dare to risk detection.

Although *GATO* is billed as a submarine simulation, it's not a simulation like Microsoft's *Flight Simulator*. You won't actually learn to operate a sub or to navigate underwater. Nevertheless, there are ample controls—depth, speed, heading, fuel, battery, torpedo, periscope—to keep your fingers busy.

You won't master *GATO* in a few days—or even weeks. The level of difficulty is set by a program parameter: At level 0, where I play, Morse-code messages are translated into English and enemy ships leave a convenient trace on the patrol chart. (Even so, my record isn't good—I complete only half of my assigned missions.) At level 9 (for Annapolis graduates, I think), you'd better know Morse code and be able to make plots of enemy activity.

This isn't a game where you can shoot at everything in sight. Successfully completing the mission is the most important goal, and accomplishing that requires the use of strategy to survive.

GATO requires a PC with color/graphics adapter, 128K of RAM, and a color monitor, or a PCjr with a color monitor. It is produced by Spectrum HoloByte, Inc. ($39.95).

A Peek At A Sequel

The *King's Quest* series of adventure games has one of the largest followings of any entertainment program for the IBM. Whenever I write about *King's Quest*, I get lots of letters—some of them quite unique. (One lady wanted to give her husband the gnome's name for his birthday.) Anyway, someone on the inside has slipped me a copy of the design specifications and some memos between the designer and programmers for *King's Quest III*, which Sierra On-Line is working on for release in late fall. I won't spoil your fun by revealing too many secrets, but I'll drop some hints of what's to come in this eagerly awaited sequel. These notes also provide some insight into how a major adventure game is carefully planned and executed by a whole team of designers, artists, and programmers. It's almost like storyboarding a film script.

From the designer's notes: "I'm going to try to make *KQ3* more difficult to solve...I'd like it to be able to do its own mapping, but Ken and Jeff will have to be talked into this...I would like to try to add more arcade-type action, but still retain the flavor of an adventure game." The notes also indicate that there will be a new routine to draw the screens because some players (including myself) are getting important clues by watching what is drawn last in a scene.

"Included in the documentation will be the magic spell book, *Sorcery of Old*." The notes mention numerous spells, including one to transform someone into a cat and another to brew up a storm. It also mentions an invisibility ointment made from toad spittle, and the new cast of characters: Medusa, a huge spider, bandits, pirates, and an abominable snowman who lives in the mountains and will drag you into his cave and devour you for dinner. The notes indicate that the best way to deal with the snowman is to use a protective spell.

Here is the designer's description of Room 25: "Ocean side. Looks like north Calif. coastline. All, part, or none of the town will be in this picture, depending on how you draw it. There will be a dock or pier going out into the ocean from the town. Later on in the game, there will be a pirate ship that is tied to the dock. The pirate ship will probably be two screens long. You can get ocean water from this room for a spell...I'm not sure yet. Maybe, we will see a pirate walking around on the deck while it is tied to the dock and his mates are in the tavern. Or maybe we'll see an old man sitting outside the tavern, or maybe a woman coming out of the store or something. Just to make the town look like it is inhabited."

The notes also indicate that Room 38 (scene 38) is inside the bandit's hideout, and that a bandit will always be there to protect a bin. What the bin contains is unclear.

If my Sierra On-Line contact, known as Deep Ego, can come up with more, I'll let you know.

* * *

Here's a tip for those of you who are running Microsoft's *Flight Simulator* on the IBM PCjr. On some TV sets the colors will fade in and out. This occurs only with version 2.11 or earlier, only on the PCjr, and only with some TV sets. Nevertheless, what looks like a hardware problem is really a bug in *Flight Simulator*. If you call your Microsoft customer service number, they have a fix. ©

A New Operating System

Computer software continually evolves, and operating systems are no different. The operating system is the core software of your computer, responsible for managing the hardware and providing routines for other programs to draw upon. The Amiga operating system, for example, contains routines that support menus, windows, memory management, and multitasking.

Most computer operating systems are stored in Read Only Memory (ROM), a permanent, nonalterable form of memory. In contrast, most application software is provided on disk, which is loaded into Random Access Memory (RAM). When updates to the software become necessary (which is almost always the case), the publisher can simply ship new disks.

The only way to upgrade software stored on ROM, though, is to pry out the original ROM chips inside the computer and replace them with new ROMs. This usually requires dealer servicing.

RAM Emulating ROM

The Amiga uses a different technique. It contains only a small amount of ROM which loads the bulk of the operating system from the *Kickstart* disk into a special area of RAM called the Writeable Control Store (WCS). Once this RAM is filled, a special switch write-protects it—effectively turning the RAM into ROM as long as the computer is turned on. The WCS cannot be corrupted by an errant application program or even a system crash.

The WCS was originally intended as a stopgap measure until the operating system could be firmed up and burned into ROM chips. But soon after the computer was introduced, Amiga recognized the value of an easily upgradable operating system and decided to stick with the WCS. One upgrade has already been released: The original version 1.0 was replaced with version 1.1 in late 1985. Version 1.1 added new features and cured legions of bugs that plagued 1.0, but it is still not perfect.

Over the past few months, Amiga has been working very hard to finish version 1.2. This upgrade was developed at first to work with the European Amiga, but includes numerous bug fixes and improvements as well. At this writing (mid-May), we have been exploring a prerelease version of 1.2, which might be available by the time you read this. Note that some features we'll describe may be changed in the final release version.

The most noticeable improvement in 1.2 is the much faster disk access due to a technique known as *caching*. A disk cache buffers disk reads so that frequently accessed areas of the disk are copied into RAM. From then on, the frequently accessed files are read from RAM rather than from the drive. It's similar to using the RAM disk, except that output is always stored on disk, not in RAM, so this technique is much safer than using a RAM disk. If the power is interrupted, you haven't lost your data.

Version 1.2 lets you choose how much memory to allocate for this disk cache—the more memory you set aside, the faster the disk access. The disk directory is also buffered, so directory-based operations such as Open requesters or an AmigaDOS DIR command work much faster. As a tradeoff, the momentary disk access that takes place when you first insert a disk lasts a little longer, since all directories and subdirectories are buffered. And, of course, there's less RAM available for applications, since the cache consumes some memory.

A Better Workbench

The Workbench is improved, too. The horizontal lines in a window's title bar have been thickened to reduce flickering in the interlaced modes. When entering text into a text gadget, you can reposition the cursor by pointing and clicking the mouse. You can use Left Amiga-V and -B as shortcuts for the affirmative and negative choices in a two-button requester. When you drag icons, you move an actual copy of the icon rather than a crossed circle. This even works with multiple selections, and is really impressive when you are dragging dozens of icons. Opening a Workbench window is no longer an excuse for a coffee break: Icons now pop up quickly, with little disk access. Any reference to the RAM: device creates an icon for the RAM disk on the Workbench screen, especially handy for one-drive systems.

A new Preferences tool lets you select serial port parameters such as data bits, stop bits, and so on, greatly simplifying the use of a serial printer or modem. There's also a toggle switch for Workbench Interlace On/Off. When Interlace is turned on, the Workbench changes to a 400-line screen with twice the vertical resolution, giving 50 lines of text.

There is a new Notepad on the Workbench disk, enhanced with an Edit menu permitting copy/cut/paste and search-and-replace. You can set up the Notepad so it calls up only one font when loaded, then bring in the fonts later from a menu if you wish. You can select either character wrap or word-wrap, and you can intermix various fonts and styles in the same note. Scroll bars let you move quickly through your text. The Notepad is now almost a complete word processor.

All in all, the new operating system is very exciting. It almost makes the Amiga a whole different machine: faster, smoother, and more reliable than ever. ©

INSIGHT: ST

Bill Wilkinson

GEM Quirks

The Atari ST is a computer with excellent hardware, but all too often problems with its system software obscure this excellence. Admittedly, most users will never actually see these problems, since software developers work hard to circumvent them. Luckily, application programmers can make a real contribution to the users' perceptions of a machine.

For example, consider the ST's floppy disk drives. In theory they are among the fastest available for any microcomputer. And indeed, when you load a program, the speed is impressive. However, when a program starts performing file input/output using ordinary record sizes, there is so much operating system overhead to overcome that the ST performance is only fair. Creating a new file with 512-byte records is only a little more than twice as fast on an ST as it is on an Atari 400/800, XL, or XE.

Possible solution: The application program can read and write very large blocks to the disk (for example, 4K or bigger), performing the file buffering itself. Suddenly the performance is quite good again. This requires a little more work on the part of the application programmer, but the net effect is pleasing for the user.

Similarly, using a hard disk on the ST is an experience not to be forgotten. For example, compiling an average-length program with *Personal Pascal* usually takes one to two minutes using floppies. When using a hard disk, those times improve to 10 or 15 seconds. That's because the hard disk port on the ST is capable of transferring more than one megabyte per second.

But something happens as the hard disk starts filling up. Access times can double before the disk is even half full. Again, there's a solution: Partition the 20-megabyte disk into four smaller, five-megabyte "logical" drives. And, since the ST uses subdirectories so successfully, this is usually a practical solution.

Gullible GEM

Perhaps the biggest problem with GEM (the Graphics Environment Manager) is that it is too gullible—tell it a lie and it believes you. Consider what happens on an Atari 400/800, XL, or XE when an Atari BASIC programmer uses a PRINT statement to display a message which is wider than the screen: The text wraps around to the next line.

When programming with GEM, the easiest way to display something on the screen is via an *alert box.* This is the small window which pops up to report errors and so forth. To display an alert box, a programmer simply defines a string of the proper form and makes an easy call to a GEM routine. But if the programmer errs when defining that string (for example, by entering too many characters or leaving out some special characters), *crash!* Time to hit the old reset button.

Now, granted, the proper form of that string is easy to validate before calling GEM, so a well-written application program will never reveal this particular problem to its user. However, this is symptomatic of much of GEM. Application programmers must do a lot of work to insure that GEM is given only legal values to work with. GEM does not seem to follow the *GIGO* rule (Garbage In, Garbage Out); with GEM it is more like *GIC* (Garbage In, Crash!). So be careful if you're writing programs on the ST. Avoid crashes by double-checking all data before calling GEM routines.

The Software Explosion

To a beginner, the ST with its GEM operating system looks complex. And, truly, there is a *lot* to learn before you can write programs which show off all the capabilities of the ST. But, despite my comments above, experienced programmers find that GEM does so much of the work for them that they can develop fairly complex programs relatively quickly. Too, the capabilities and accessibility of higher-level languages for the ST (such as C, Pascal, and Modula-2) have made programmers more productive. As a result, there is arguably more software available for the ST, *at this point in its life,* than for any previous computer at a comparable point in its life.

For instance, one year after the Macintosh was introduced, it had far fewer programs available than the ST has about one year after its introduction. Not only that, the ST programs tend to be considerably less expensive than their Macintosh counterparts.

One of the reasons so much software is appearing is that the cost of developing for an ST is relatively low. A part-time ST programmer can have a full-blown ST development system for not much over $2,000 (including hard disk, printer, color and monochrome monitors, development software, and so forth). In the early days of the Mac, $10,000 was more the order of the day, so development tended to be restricted to established software companies.

The flip side of this coin is that the quantity of *high-quality* software for the ST is certainly *not* greater than what was available for the Macintosh. Since most early Mac developers were major software companies, their quality standards were generally higher than that of part-time hackers.

Bottom line: Try to see a demo of any ST software you are planning to purchase. There are a lot of excellent ST programs, but there are also some turkeys. ©

Programming the TI

C Regena

An Amortization Schedule

Interest rates have been plunging lately, and it seems like home mortgages and refinancing are very popular topics for newspaper articles. Recently I was reading a question-and-answer article in which the reader asked for a program for his home computer to print an amortization schedule for a home mortgage. The columnist suggested a particular program which was easy to use and costs only $99. I couldn't believe someone would spend $99 for a program that uses one or two basic computations! So, for the price of this magazine, here is such a program: "Loan Amortization."

It's certainly easy to use. Just enter the amount of money you want to borrow, omitting the dollar sign and comma (i.e., type 50000 instead of $50,000). Next, enter the interest rate, such as 13 for 13 percent or 9.5 for nine and a half percent. Finally, enter the number of years for the loan. Most loans are for a certain number of whole years, such as 25 or 30, so this program is based on 12 monthly payments per year rather than calculating a number of months. The program then tells you what your monthly payment will be. (Of course, this figure doesn't include property taxes, insurance, or condominium fees.)

You may then choose to see the amortization schedule on the screen or print out a paper copy. If you have a printer, be sure to use the correct printer configuration in line 710, the OPEN statement. If you don't want to see the amortization schedule, you may calculate another loan or end the program.

Converting Math To BASIC

Among other things, Loan Amortization demonstrates how easy it can be to convert a mathematical formula into a BASIC program. Any ordinary formula can be converted by using the + and − signs for addition and subtraction, the * sign for multiplication, / for division, and sets of parentheses where necessary to group mathematical operations.

Use PRINT and INPUT statements to prompt numbers from the user. You may want to use some IF-THEN statements to make sure the INPUT values are within reasonable limits for the formula. In Loan Amortization, all numbers entered must be positive. The amount of the loan has to be six digits or less (not counting the cents) to help limit the printing variables. The number of years is from 1 to 50.

Once your program has all the numbers it needs, calculate the formula and PRINT the answer. The computer, of course, is ideal for handling repetitious calculations, such as this amortization schedule.

Any economics book has formulas for various calculations involving money—savings accounts, sinking fund deposits, present worth factors, and so forth. In this case, to find the monthly payment I used the capital recovery factor formula:

$$I(1+I)^N/(1+I)^N-1$$

where I = interest and N = the number of payments. To make it easier to type the program without errors, I used the variable D for interest, since the letter I can be confused with the numeral 1. Then the program converts the percentage to a monthly decimal, J=D/1200. The factor with the exponent is used twice, so I calculated it as F in line 490. Line 500 then calculates the capital recovery factor, CRF.

How To Pause Printing

The FOR-NEXT loop in lines 800–1050 prints the amortization schedule with the monthly payment PAY. Part of the payment goes to principal (the variable PR), and part is interest (the variable II). The balance is the original principal minus the principal part of the payment, P. Lines 1060–1200 calculate and print the last payment, which may be slightly different than the regular monthly payment because of rounding to the cent.

The printing on the screen includes only the month number, principal and interest, then balance. To pause the printing while it is scrolling, hold down any key. When you release the key, the schedule will continue. To make this work, lines 1010–1040 scan the keyboard in each loop. If a key is not pressed, the program goes to the next calculation. You may want to print different items or adjust the printing to better suit your needs.

All of the PRINT # statements send text to the printer. The variables L1, L2, and L3 are lengths used in the TAB functions to line up the columns. The variable R holds the user's choice: 1, 2, 3, or 4. If the choice is 1, the program skips all the statements that pertain to the printer.

The subroutine in lines 1250–1330 converts a number in the variable A to a string so that a number can be written in money form with two decimal places (using zeros where necessary). The numbers are rounded to the nearest cent.

If you have TI Extended BASIC or are converting this program to another version of BASIC, PRINT USING would be easier to use than this subroutine. For example, PRINT USING ####.## will round a number to two decimal places and will also right-justify numbers for printing straight columns.

Loan Amortization

```
100 REM  AMORTIZATION
110 CALL CLEAR
120 PRINT "THIS PROGRAM WIL
    L CALCULATE"
130 PRINT "A MONTHLY PAYMEN
    T FOR A"
```

```
140 PRINT "GIVEN PRINCIPAL
    BORROWED"
150 PRINT "AT A CERTAIN INT
    EREST RATE."
160 PRINT ::"ENTER AMOUNT B
    ORROWED."
170 INPUT PP
180 IF PP>0 THEN 210
190 PRINT "PLEASE ENTER AMO
    UNT > 0"
200 GOTO 160
210 IF PP<999999.01 THEN 25
    0
220 PRINT "THIS PROGRAM IS
    FOR LOANS"
230 PRINT "LESS THAN $99999
    9."
240 GOTO 160
250 PRINT ::"ENTER INTEREST
     RATE IN %."
260 INPUT O
270 IF O>=0 THEN 300
280 PRINT "PLEASE USE POSIT
    IVE PERCENT."
290 GOTO 250
300 PRINT ::"ENTER NUMBER O
    F YEARS FOR"
310 PRINT "LOAN."
320 INPUT Y
330 IF (Y>=1)+(Y<51)=-2 THE
    N 370
340 PRINT "THIS PROGRAM IS
    FOR LOANS"
350 PRINT "FROM 1 YEAR TO 5
    0 YEARS."
360 GOTO 300
370 IF Y=INT(Y)THEN 400
380 PRINT "NO FRACTIONAL YE
    ARS PLEASE."
390 GOTO 300
400 CALL CLEAR
410 PRINT "AMOUNT BORROWED:
     ";PP
420 PRINT :"INTEREST RATE:
    ";O;"PERCENT"
430 J=O/1200
440 PRINT "TIME IN YEARS: "
    ;Y
450 N=12*Y
460 IF O<>0 THEN 490
470 CRF=1/N
480 GOTO 510
490 F=(1+J)^N
500 CRF=J*F/(F-1)
510 PRINT STR$(N);" MONTHLY
     PAYMENTS"
520 A=PP*CRF
530 GOSUB 1250
540 PAY=A
550 PAY$=A$
560 PRINT "MONTHLY PAYMENT
    = $";A$
570 PRINT ::"PRINT AMORTIZA
    TION?"
580 PRINT :"1 YES, ON SCREE
    N"
590 PRINT "2 YES, ON PRINTE
    R"
600 PRINT "3 NO, TRY ANOTHE
    R LOAN"
610 PRINT "4 NO, END PROGRA
    M"
620 CALL KEY(0,K,S)
630 IF (K<49)+(K>52)THEN 62
    0
640 CALL CLEAR
650 R=K-48
660 ON R GOTO 670,670,110,1
    340
670 A=PP
680 GOSUB 1250
690 P=A
700 IF R=1 THEN 750
710 OPEN #1:"RS232.BA=600"
720 PRINT #1:"AMOUNT BORROW
    ED:  $";A$
730 PRINT #1:"INTEREST RATE
    : ";O;"PERCENT"
740 PRINT #1:::"MONTH    PA
    YMENT";TAB(30);"PRINCIP
    AL";TAB(50);"INTEREST";
    TAB(65);"BALANCE":::
750 PRINT "TO PAUSE PRINTIN
    G, HOLD ANY KEY DOWN.
    RELEASE KEY TO   CONTIN
    UE.":::
760 PRINT "AMOUNT BORROWED:
     ";A$
770 PRINT "INTEREST RATE: "
    ;O
780 PRINT :"MONTHLY PAYMENT
    : $";PAY$
790 PRINT :"     PRINCIPAL
       INTEREST";TAB(12);"B
    ALANCE"::
800 FOR M=1 TO N-1
810 M$="  "&STR$(M)
820 M$=SEG$(M$,LEN(M$)-2,3)
830 A=J*P
840 GOSUB 1250
850 II$=A$
860 II=A
870 L2=6-L
880 A=PAY-I1
890 GOSUB 1250
900 PR$=A$
910 PR=A
920 L1=6-L
930 A=P-PR
940 GOSUB 1250
950 P=A
960 P$=A$
970 L3=6-L
980 PRINT M$;"  ";PR$;TAB(1
    8+L2);II$:TAB(10+L3);P$
990 IF R=1 THEN 1010
1000 PRINT #1:" ";M$;TAB(11
     );PAY$;TAB(31+L1);PR$;
     TAB(51+L2);II$;TAB(65+
     L3);P$
1010 CALL KEY(0,K,S)
1020 IF S<1 THEN 1050
1030 CALL KEY(0,K,S)
1040 IF S<0 THEN 1030
1050 NEXT M
1060 M$="  "&STR$(M)
1070 M$=SEG$(M$,LEN(M$)-2,3)
1080 A=J*P
1090 GOSUB 1250
1100 II$=A$
1110 I1=A
1120 L2=6-L
1130 A=II+P
1140 GOSUB 1250
1150 PAY=A
1160 PAY$=A$
1170 L1=6-L
1180 PRINT M$;"  ";P$;TAB(1
     8+L2);II$:TAB(15);"0"
1190 IF R=1 THEN 1220
1200 PRINT #1:" ";M$;TAB(11
     );PAY$;TAB(31+L1);P$;T
     AB(51+L2);II$;TAB(68);
     "0"
1210 CLOSE #1
1220 PRINT ::"PRESS A KEY"
1230 CALL KEY(0,K,S)
1240 IF S=0 THEN 1230 ELSE
     570
1250 A=INT(A*100+.5)
1260 A$=STR$(A)
1270 L=LEN(A$)
1280 IF L>=2 THEN 1310
1290 A$="0"&A$
1300 L=2
1310 A$=SEG$(A$,1,L-2)&"."&
     SEG$(A$,L-1,2)
1320 A=VAL(A$)
1330 RETURN
1340 END
```

©

ATTENTION T.I. 99/4A OWNERS

- Diskettes - 59¢ each! Your choice SS or DD
- 512K Now Available for the 99/4A!
- 99/8 Level 4 Computer Upgrade Now Available
- Over 1500 Herdwere end Software Accessories at Similar Savings

THE WORLD'S LARGEST COMPUTER ASSISTANCE GROUP

Now serving over 35,000 members worldwide with the best in technicel essistence, service, end products for the Texes Instrument 99/4A Computer

To become e member end receive newsletters, cetelog, technicel essistence end membership peckege, send $10 00 for e ONE Yeer Membership to.

99/4A National Assistance Group
National Headquarters
P.O. Box 290812
Ft. Lauderdele, Floride 33329
Attention Membership Division
For Further Information Call 24 Hours
(305) 583-0467

STATE-OF-THE-ART MAGNETIC MEDIA

5¼" DISKETTES

- With Hub Rings
- Write Protect Tebe
- Envelopes
- User ID Labels
- In Fectory Seelsd Poly Packs of 10

(YOU GET EVERYTHING BUT THE eOX)
Prices ere per Disk

QTY	50	100	500	1000
SSDD	.59	.53	.50	.47
DSDD	.62	.58	.55	.52

Library Case Holds 15 Diskettes ..... Only **$1.00** plus 50¢ S&H
The 100 File ..... Only **$10.95** plus $2.00 S&H
100% ERROR FREE — LIFETIME WARRANTY
Min order $25 00 Add 10% for less than 50 diskettes **Shipping end Hendling** $4 00 per 100 diskettes Reduced shipping for larger quantities C O D edd $4 00 Cash or certified check
Continental USA C.O.D.

Precision Data Products
P O Box 8367 Grand Rapids MI 49518
(616) 452-3457 • Michigan 1-800-632-2468
Outside Michigan 1-800-258-0028

In The Car — **Peace of Mind** Only $3.00 a year!* — In The Kitchen

HALONITE

10 Year Manufacturer's Gucrantee with a price under $30 (1 lb Unit).
(*each guarantee year only $3 00)
Halonite-A patented blend of 1301/1211 Expelled as a vapor — Safe around food Will not harm computers and penpherals —Safe for ALL fires —

To Order By Phone
(Visa & MasterCard Only)
1-800-638-2000
In Maryland
1-800-831-4300

Larger Sizes Write

For Info & Orders
G*Ltd.
P O Box 215
Bath, PA 18014
(in PA add 6% Tax)

Item #AH10 $29 95 (+3 00 Shipping & Handling) 2½ Dia x 7⅞ H x 5½ W — 1 lb Charge

Item #AH15 $39 95 (+3 50 Shipping & Handling) 2⅞ Dia x 9¼ H x 5⅞ W — 1 5 lb Charge

Item #AH22 $59 95 (+3 50 Shipping & Handling) 3⅝ Dia x 10¼ H x 6 W — 2 2 lb Charge

News & Products

Penguin Software Announces Price Drop

Penguin Software has announced an across-the-board price drop for its software line. All programs in its COMPREHEND Interactive Novel series will be $17.95 for 5¼-inch disks (Apple, Commodore 64, IBM) and $19.95 for 3½-inch disks (Atari ST, Macintosh, and Amiga). This line includes such titles as *Crimson Crown, Oo-Topos,* and *Transylvania.* Suggested retail price for *Graphics Magician* and *Complete Graphics System* will be $39.95 ($49.95 for Macintosh version).

Other graphics utilities, such as *Paper Graphics, Transitions,* and *Cat Graphics* will be priced at $19.95. *Graphics Magician Junior* (Apple and Commodore) will be $19.95. *Disk Repair Kit* will be $12.95. In the Home series, *Home Data Manager* will be priced at $24.95, and *Home Connection* (with $15 free CompuServe time) will be $29.95. Also, some backlist titles will be available for $8.95.

Penguin Software, 2600 Keslinger Rd., P.O. Box 311, Geneva, IL 60134.

Circle Reader Service Number 220.

New Reading, Social Studies Software

CBS Interactive Learning has introduced *The Novel Approach: Lord of the Flies,* the first title in the Novel Approach computer software series developed by Media Basics for Apple, IBM, and Commodore eight-bit systems. Each program in the series focuses on a popular literary classic frequently studied in junior and senior high school. Designed to help students develop or maintain interest in reading and to build critical reading skills, each title in the Novel Approach can be used as a springboard for classroom discussion or independent study. Four additional Novel Approach titles are planned for release in the fall of 1986. They are *Animal Farm* by George Orwell, *A Tale of Two Cities* by Charles Dickens, *The Call of the Wild* by Jack London, and *Romeo and Juliet* by William Shakespeare.

The Novel Approach series motivates students to read by enhancing their understanding and appreciation of literature. Each program helps students focus on character motivation, plot development, symbolism, narrative techniques, and vocabulary. Rather than replacing the reading of the book itself, the programs are meant to be used before, during, and after reading. Each includes three separate learning activities: The Discoverer, designed to pique interest before reading; The Explorer, a self-paced series of questions and answers that enhance understanding; and The Master, designed to test students' knowledge of the story after it has been read.

Built into each program in the Novel Approach series is a comprehensive reference guide, *The Book Scanner.* It provides background information on each book, a profile of the author, and an annotated bibliography of related books. Errors are tracked, and corrections with explanations are provided.

The Novel Approach: Lord of the Flies comes with a program guide, teacher's guide, and backup disk. It is available for the Apple II series (48K RAM minimum), Commodore 64, and IBM-PC and PCjr with 128K RAM and graphics board for $59.95.

CBS also has introduced *Continents and Countries,* a program for use within the social studies curriculum in grades 5–12. Designed by Neosoft, the program helps students build and test their knowledge of the nations and peoples of the world through self-paced learning activities. Its database covers over 140 countries and includes facts on each country's major religion, language, per capita income, land area, form of government, and population. *Continents and Countries,* available for

C-64™ • VIC™ • SX-64™ • C-128™ • Plus 4™ • C-16™ • B-128™ • PET™ • CBM™ • LCD™

One disk. 25 business programs, $19.95

The Intelligent Software Package is the one product for your Commodore that can take care of all your data processing needs.

Customers write: *". . . accolades for the authors. This is as slick a deal as I have seen and more than adequate for all except fancy presentations. The best thing is the ease of use . . ."*

"I have come to consider these programs among the most valuable pieces of software I own."

There are no hidden fees for shipping or documentation, and no clubs to join. The package is not public domain software, and is sold only direct to customers by mail: it supports all available printers, and will run on any Commodore computer (except Amiga) with a minimum of 10k RAM, including the C-128 in C-128 mode.

What you get when you order the Package:

Database—A complete database manager. All fields completely user-definable. Can be used for any number of tasks, including accounting, checkbook and tax records, mailing lists, inventory control, catalog maintenance, or as an electronic rolodex. A customer writes: *"I am especially impressed with Database, and have used it to replace a half-dozen other 'database'-type programs I had been using."*

Word Processor—A full-featured menu-driven word processor. Allows full control over margins, spacing, paging, indentation, and justification. *"Highly recommended."*—Midnite Software Gazette. *"Provides good basic features."*—Compute's Gazette

Copycalc—An electronic spreadsheet. *"Excellent program for budgeting, estimating, or any math-oriented use . . . well worth the money. Highly recommended."*—Midnite Software Gazette.

ReportGen—creates form letters, mailing lables, etc.

ReportMerge—creates statements invoices.

Baseball Stats—compiles team batting statistics.

Index—indexes W/P's text files.

Wordcount—counts words in a text file.

WPConvert—converts files to other WP formats.

DBMerge—facilitates relational D/B applications.

DBStat, DBStat2—analyze D/B files.

ASCII—converts text files into program files.

Checkbook—reconciles checkbook.

Inventory—Maintains inventory records.

Paper Route—A/R for paper route.

Loan Analysis—computes finance terms, prints schedules.

Breakeven—computes breakeven analysis.

Depreciation—creates depreciation schedules.

Labeler—creates labels.

File Copier—copies sequential, program files.

Correlation—calculates statistical correlation.

Also other Database and Word Processor utilities.

To order, send name, address, and $19.95 to address below. Please specify regular (1541/1571/2040/4040/2031) disk, 8050 disk, or cassette (cassette not available for Plus 4 or C-16). Add $3 for credit card or COD orders. Calif. residents add 6%. No personal checks from outside USA. A sampling of program output is available for $1. **Tear this ad out and keep it handy!**

Intelligent Software
Quality Software since 1982

Box A Dept. T-8
San Anselmo, CA 94960
(415) 457-6153

the Apple II series with 48K RAM minimum, has a suggested retail price of $49.95.
CBS Interactive Learning, One Fawcett Pl., Greenwich, CT 06836.
Circle Reader Service Number 221.

Baseball And Bridge For Apple

Random House has announced Apple II conversion of two programs. *APBA Major League Baseball* uses actual statistics from the 1984 or 1985 baseball season and lets users start their own leagues, draft teams from a list of 676 big-league players, or play with the actual rosters for all 26 teams from each season. The program is now available for Apple IIe and IIc with 128K, 80-column card, and two disk drives for $59.95.

Tournament Bridge offers competition and practice for the serious bridge player. It is available for the Apple II+, IIe, and IIc for $49.95. Random House also is developing a word processor for Apple II computers that uses a Macintosh- style user interface. A fall release is planned.
Random House, Electronic Publishing Division, 201 E. 50th St., New York, NY 10022.
Circle Reader Service Number 222.

Foreign Language Detective Game

Gessler Educational Software has announced French, Spanish, and German versions of Tom Snyder's bestselling program *Snooper Troops*. The program helps children develop their foreign language vocabularies and reasoning skills by having them take notes, draw maps, and organize information.

The object of the Spanish and German versions is for the player to determine who committed a crime in the old mansion and why. The player must question the suspects and remember each correct password and clue in order to solve the mystery. Available for the Commodore 64, the Spanish and German versions retail for $39.95. In the French version, available for Apple II+, IIe, and IIc at $49.95, the player's mission is to find the villain who fled with Lily the Dolphin.
Gessler Educational Software, 900 Broadway, New York, NY 10003.
Circle Reader Service Number 223.

More New Releases From The U.K.

Firebird Licensees, a British software licensing company which made a nice entry into the U.S. market with *Elite*, recently introduced several new products.

The Pawn is a rich text-and-graphics adventure previously available for the Atari ST, but now shipping for the Commodore 64 and 128 (in native 128 mode). Set in the mythical world of Kerovnia, the game provides the player with an intricate network of plots and subplots with many objectives.

New members of the Firebird "flippy" family (disks with one program on each side) are *Battle of Britain/Battle of Midway* (Commodore 64, $19.95), strategy/war games that break out into arcade-style games at certain points in the action; *Iwo Jima/Falklands '82* (Commodore 64, $19.95); and *Chimera/Mercenary* (Atari 800/130, $19.95).
Firebird Licensees, P.O. Box 49, Ramsey, NJ 07446.
Circle Reader Service Number 224.

ST, Amiga Programs

Classic Image is releasing two programs each for the Atari ST-series computers and the Commodore Amiga.

Disk Library is a tool for keeping track of files on your disks. Files, folders, and subdirectories can be categorized and cross-referenced. Lists of files and folders can be displayed on the screen or dumped to a printer. Disks can be searched by any category, and the entire library is automatically updated as new disks are added. *Disk Library* works with single- or multiple-drive systems and is available for both the ST and Amiga for $49.95.

Diablo is an original game that combines animation with strategy. The screen is filled with mazelike tracks that disappear in sections as a ball rolls over them. The player's goal is to keep the ball rolling as long as possible without running out of track. Versions for the ST and Amiga retail for $29.95 each.
Classic Image, 510 Rhode Island Ave., Cherry Hill, NJ 08002.
Circle Reader Service Number 225.

Turbocharged Amiga

Computer System Associates has introduced a series of add-on circuit boards that modify a Commodore Amiga for high-speed operation using the Motorola 32-bit 68020 microprocessor.

A specially modified Turbo-Amiga runs at a CPU (central processing unit) clock speed of 14 megahertz, contains up to 2.5 megabytes of 32-bit memory, and can accept an optional Motorola 68881 math coprocessor. The 68020 modification alone increases overall performance by about 60 percent. By adding 512K of 32-bit memory, performance increases about 140 percent. Applications which use intensive floating-point math can run up to 40 times faster. Complete Turbo-Amiga systems start at $4,980.
Computer System Associates, 7564 Trade St., San Diego, CA 92121.
Circle Reader Service Number 226.

ST, Amiga Golf Game

Accolade has announced that versions of its golf-simulation game will be available this summer for the Atari ST-series and Commodore Amiga computers.

Mean 18: Ultimate Golf uses 3-D animation to simulate golfing on three famous courses—Pebble Beach, St. Andrews, and Augusta National. In addition, you can construct your own courses. A bird's-eye view shows the position of your ball after each shot. Different levels of difficultly accommodate all kinds of players. The ST and Amiga versions of *Mean 18* will retail for $49.95 each.
Accolade, 20833 Stevens Creek Blvd., Cupertino, CA 95014.
Circle Reader Service Number 227.

Recreational Software

Baudville is releasing three new home and educational programs for the Commodore 64, Atari 400/800/XL/XE, Apple II series, IBM PC and compatibles, Atari ST series, Commodore Amiga, and Apple Macintosh.

Video Vegas recreates authentic casino games such as blackjack, draw poker, keno, and slot machines. *Guitar Wizard* helps both novice and experienced musicians learn and analyze scales, chords, and tunings for all kinds of fretted string instruments. *Ted Bear's Rainy Day Games* is a three-in-one card game package for youngsters. It contains computer versions of concentration, old maid, and go fish.

All of the programs are scheduled for release this fall at prices ranging from $29.95 to $34.95.
Baudville, 1001 Medical Park Dr., SE, Grand Rapids, MI 49506.
Circle Reader Service Number 228.

Color Printer For Amiga, ST

Okidata has released adapters to make its Okimate 20 color thermal-transfer printer work with the Commodore Amiga and Atari ST-series computers.

The Plug 'N Print Modules for the Amiga and ST include a cable, cartridge ribbons, paper, and instructions. The Okimate 20 has a 24-element thermal printhead that reproduces more than

ATARI 130XE

ATARI 130XE Super Computer Package

130XE Computer — 1027 Printer
1050 Disk Drive — Atariwriter +

Call for individual & super package price

ATARI PRINTER INTERFACES

Uprint A . . . $4.95 MPP 1150 . . . $4.95

Supra 1000E Modem 44.95
Atari XM-301 Modem 39.95

ATARI 130-XE SUPER PRINTER PACKAGES

NX-10 Printer & U-Print A 299
Panasonic 1091 & U-Print A 299

Super Printer Packages have no extra shipping charges or credit card surcharges when shipped in Continental USA

ATARI 130XE SOFTWARE

BRODERBUND

Print Shop	28.95
Karateka	20.95
Print Shop Graph I II or III	19.95
Print Shop Comp	27.95

INFOCOM

See Commodore 64 section for items and prices

ELECTRONIC ARTS

Archon II	24.95
Archon	19.95
Seven Cit of Gold	24.95
Skytox	24.95
Pinball Const	19.95
One on One	24.95
Super Boulder Dash	19.95
Chessmaster 2000	27.95
Racing Destruction	24.95

MICROPROSE

Silent Service	23.95
Gunship	23.95
Accrojet	23.95
F 15 Strike Eagle	23.95
Kennedy Approach	23.95
Conflict/Vietnam	27.95

SSI

See Commodore 64 section for items and prices

MISCELLANEOUS 130XE

Mind Pursuit	23.95
Never Ending Story	23.95
Foobliczky	27.95
Syncalc	32.95
Typesetter	27.95
Ultima III	37.95
Ultima IV	41.95
Raid Over Moscow	27.95
Fight Night	23.95
Hardball	23.95
Flight Simulator II	34.95
Alternate Reality	27.95
Fleet System II	49.95
Page Designer	23.95
Megafont II	19.95
Rubber Stamp	23.95
Halley Project	23.95
Synfile	32.95
Beachead II	27.95
Music Studio	20.95
Wizard s Crown	27.95
Gettysburg	39.95
Island Caper	23.95
Basic XE	52.95
Action	49.95
MAC 65XL	49.95
Paper Clip XE	39.95

ATARI 520ST*

Atari 520ST-RGB System . . . . . Call
Atari 520ST-Monochrome Sys . . . Call
SF314DS/DD1 Megabyte Disk Dr . . . 209

We warranty all 520ST computers purchased from ComputAbility for ninety days

***Please call for stock availability on Atari ST products before ordering by mail.**

MISCELLANEOUS ST

Galo	34.95
Kings Quest II	34.95
ST Talk	17.95
H&D Base	69.95
Typesetter ST	27.95
Habadex Phone	39.95
Word Invaders	24.95
Mean 18 Golf	34.95
Financial Cookbook	34.95
Braltaccus	34.95
Michtron Utilities	41.95
Exploding Fist	34.95
Black Cauldron	27.95
Printmaster ST	27.95
PC Intercom	89.95
Final Word	94.95
Hex	27.95
Mudpies	27.95
Sundog	27.95
Flipside	27.95
Softspool	27.95
VIP Professional	119.95
Ultima II	39.95
Perry Mason	34.95
Degas	27.95
Farenheit 451	34.95
Personal Pascal	54.95
Checkminder	54.95
Printmaster	27.95
Art Gallery I	20.95
Compubridge	20.95
General Ledger	84.95
Major Motion	27.95
Pawn	27.95
DFT	34.95
Universe II	49.95
Macrodesk	20.95
Rubber Stamp	27.95
Kissed	34.95
Cards	27.95
Rouge	27.95
Amazon	34.95
OOS Shell	34.95
Kissed	27.95
9 Princess/Amber	34.95
Hacker	29.95
Dragonworld	34.95
Treasure Island	27.95
Wizard of Oz	27.95
Transylvannia	27.95
Borrowed Time	34.95
Mi Term	34.95
Regent Word	34.95
Regent Spell	34.95
Regent Base	Call
Goldrunner	27.95
Time Bandit	27.95
Zoomracks	49.95
Easy Draw	99.95
Mindshadow	34.95
Phantasie	27.95
Hippopotamus	Call
Supra Hard Drive	Call
Supra 1200 ST Modem	159
OMI 1200 ST Modem	159
PC Board Designer	Call
Infocom	See IBM
Apshai Trilogy	27.95
N Vision	27.95
OB Man	69.95
Small Bus Pkg	84.95
Cornerman	34.95
ALT	34.95
M Copy	54.95
Winter Games	27.95
Winnie The Pooh	19.95
Silent Service	Call
Flight Simulator II	Call
Music Studio	39.95
Phantasie	27.95
1 Megabyte Upgrade	175
Extended Warranty	Call

AMIGA

Call for Hardware and Add on Peripherals prices

AMIGA SOFTWARE

Hacker	29.95
Mindshadow	29.95
Masterlype	27.95
VIP Professional	139.95
Transylvannia	27.95
Crimson Crown	27.95
Racter	29.95
Analyze	69.95
On-line	49.95
Artic Fox	29.95
Braltaccus	34.95
Adventure Const	29.95
Deluxe Video	Call
Deluxe Paint	69.95
Infocom	See IBM
Monkey Business	19.95
Keyboard Cadet	27.95
Rouge	27.95
Apshai Trilogy	27.95
Delta Patrol	19.95
Maxicom	39.95
Maxidesk	49.95
Maxiplan	119.95
One on One	29.95
7 Cities/Gold	29.95
Skytox	29.95
Marble Madness	29.95
Return/Atlantis	29.95
Archon	29.95
Aegis Animator	99.95
Exploding Fist	27.95
Deluxe Print	69.95
Music Studio	39.95
Borrowed Time	29.95
Financial Cookbook	34.95
Scribble	69.95
Halley s Project	29.95
Kid Talk	39.95
Speller Bee	39.95
Gizmo	34.95

APPLE

SOFTWARE

BRODERBUND

Print Shop	39.95
Print Shop Graphics I II or III	17.95
Print Shop Comp	27.95
Karateka	21.95

ELECTRONIC ARTS

Bard s Tale	29.95
Auto Duel	34.95
Skytox	27.95
Lords of Conquest	27.95
Ultima IV	39.95
Moebius	39.95
Amnesia	Call

EPYX

Winter Games	24.95
Summer Games II	24.95

INFOCOM

See IBM section for items and prices

MICROPROSE

See Atari 130XE section for items and prices

SIR-TECH

Wizardry/Diam	23.95
Wizardry/Legacy	27.95
Wizardry/Proving	33.95

SSI

See Commodore 64 section for items & prices

APPLE MISCELLANEOUS

Gamemaker	34.95
Hardball	24.95
Sundog	27.95
Newsroom	39.95
Clip Art Vol I	20.95
Clip Art Vol II	27.95
Gato	27.95
Kung Fu Master	25.95
Karate Champ	25.95
Strip Poker	23.95
Math Blaster	34.95
Fight Night	24.95
Phantasie II	27.95
Rambo	27.95
Amer Challenge	27.95
Crossword Magic	34.95

IBM PC

Corona PC-400 Compatible . . Call
Corona Portable PC Compatible . Call

IBM PC SOFTWARE

BRODERBUND

Print Shop	39.95
Print Shop Graph I	27.95
Bank Street Writer	49.95
Ancient Art of War	29.95

MICROPROSE

F 15 Strike Eagle	23.95
Kennedy Approach	27.95
Acrojet	27.95
Silent Service	27.95

IBM MISCELLANEOUS

Mean 18	34.95
Gato	27.95
Wizardry	39.95
Strip Poker	27.95
Jet	34.95
Newsroom	39.95
Superbowl Sunday	24.95
Home Pak	34.95
Alter Ego	34.95
Hacker	29.95
Borrowed Time	27.95
Isgur Portfolio	164.95
Typing Tutor	34.95

INFOCOM

Deadline	34.95
Starcross	34.95
Zork I	27.95
Zork II or III	29.95
Witness	27.95
Suspended	34.95
Planetfall	27.95
Sorcerer	29.95
Seastalker	27.95
Cutthroats	27.95
Hitchiker	27.95
Suspect	29.95
Wishbringer	27.95
Infidel	29.95
Enchanter	27.95
Spellbreaker	34.95
Mind Forever	34.95
Ballyhoo	27.95

SIERRA

King s Quest	34.95
King s Quest II	34.95
Ultima II	39.95

COMMODORE

COMMODORE 128

C 128 Computer	Call	1670 Modem	159
1571 Disk Drive	Call	1350 Mouse	42.95
1902 Monitor	Call	1750 512K Expander	Call

COMMODORE 128 SOFTWARE

Swiftcalc w/Side	49.95	Superscript 128	59.95
Wordwriter + Spell	49.95	Perfect Writer	49.95
Data Manager II	49.95	Sylvia Porter-128	49.95
Fleet System III	54.95	Matrix	39.95
Mach 128	34.95	Perfect Filer	49.95
Superbase 128	69.95	Paperback 128	34.95
King s Quest II	34.95	Vizwrite 128	Call
Home Pak	34.95	Vizistar 128	Call

COMMODORE 64

NEW 64C Computer	Call	1660 Modem	49.95
C 64 Computer	Call	1670 Modem	159
1541 Disk Drive	Call	1802 Monitor	Call

COMMODORE 64/128 SUPER PRINTER PKGS.

NX-10 & Xetec Supergraphic 309
Panasonic 1091 & Xetec Supergraphic 309
Legend 1080 & Xetec Supergraphic 269

Super Printer Packages have no added shipping or charge card surcharges when shipped in Continental USA

While Supplies Last

B.I. 80 Column Card 59.95

XETEC Super Graphic 69.95
Cardco G-Wiz 54.95

COMMODORE 64 SOFTWARE

ACCESS

Mach V Cart	21.95
Leader Board	27.95
Tenth Frame	27.95

SSI

Battle of Antietam	32.95
Fighter Command (No Atari)	37.95
Panzer Grenidier	24.95
USAAF	37.95
Breakthrough/ Ardennes	37.95
Kampgruppe	37.95
Phantasie II (No Atari)	27.95
Broadsides	24.95
Carrier Force	37.95
Comp Ambush	37.95
Field of Fire (No Apple)	24.95
Gemstone Warrior	21.95
Imp Galactum	24.95
Computer Baseball	24.95
Comp Quarterback	24.95
Wizards Crown (No Atari)	27.95
Rings of Zilfin (No Atari)	27.95
Battle Group (No Atari)	39.95
Gettysburg	39.95

MICROPROSE

See Atari 130XE section for items and prices

ABACUS

Call for items and Prices

INFOCOM

Zork I	24.95
Zork II or III	27.95
Deadline	29.95
Starcross	29.95
Witness	24.95
Planetfall	24.95
Hitchiker	24.95
Enchanter	24.95
Cutthroats	24.95
Sorcerer	24.95
Spellbreaker	29.95
Ballyhoo	27.95

ELECTRONIC ARTS

Adv Construction	29.95
Mail Order Monster	24.95
Racing Destruction	24.95
Ultima III	37.95
Ultima IV	41.95
Bard s Tale	27.95
Lords of Conquest	27.95
Chessmaster 2000	27.95
Touchdown Fball	23.95
Ultimate Wizard	24.95
Super Boulderdash	19.95
Mind Mirror	24.95

See Atari 130XE section for rest of items & prices

EPYX

Winter Games	24.95
Apshai Trilogy	24.95
Fast Load Cart	24.95
Vorpal Utility	22.95
Multiplan	44.95
Movie Monster	24.95
World Karate	19.95
Street Sports	24.95
Super Cycle	24.95

MISCELLANEOUS COMMODORE 64

Freeze Frame	34.95
Print Shop	28.95
Cal-Kit	34.95
Superbase 64	47.95
Karateka	20.95
Hacker	20.95
Gamemaker	27.95
Ultima II	37.95
Karate Champ	25.95
Sticks of Death	20.95
Kung Fu Master	25.95
Paper Clip/Spell	54.95
Consultant	41.95
Spy vs Spy Vol II	23.95
Internat l Hockey	19.95
Prt Shop Compan	27.95
Prt Shop Graphics	19.95
Galo	20.95
Jet	29.95
Fleet System II	44.95
Printmaster	24.95
Newsroom	34.95
Fontmaster II	34.95
Sports Lib Vol I	19.95
Geos	39.95
Hidden Assets	Call
Adv Music System	54.95
Spitfire 40	23.95

We stock hundreds of programs for the Apple, Atari, C-64 and IBM. If you don't see it here, don't hesitate to call

GENERAL HARDWARE

Star Micronics Inc — The Power Behind the Printed Word

NX-10	**Call**
SG-10	**Call**
SG-15	369
SD-10	339
SD-15	449
SR-10	489
SR-15	Call
NL-10	Call
NL-10C	Call

PRINTERS

Panasonic 1091	299
Legend 808	169
Legend 1080	209
Powertype	309
Juki 551D	389
Epson	Call
Panasonic 1080	215

PRINTER BUFFERS

Microfazer	From 169
U-Buff 16K	79.95
U-Buff 64K	99.95

MODEMS

Volksmodem 1200	169
Prometheus 1200	299
Maxwell 1200	Call
Maxwell 2400	Call
Prometheus 2400	Call

MONITORS

Thommson RGB	289
Commodore 1802	Call
Commodore 1902A	289
Gold Star	Call
Amdek	Call
Panasonic 1300DT	279
NEC 1225	139

EST. 1982

ComputAbility™

PO Box 17882 Milwaukee WI 53217
ORDER LINES OPEN
Mon Fri 11 a m 7 p m CST
Sat 12 p m 5 p m CST

NO SURCHARGE FOR MASTERCARD & VISA

To Order Call Toll Free
800-558-0003

For Technical Info., Order Inquiries, or for Wisc. Orders
414-351-2007

ORDERING INFORMATION Please specify system For fast delivery send cashier s check or money order Personal and company checks allow 14 business days to clear School PO s welcome **C O D** charges are $3.00 In Continental U S A include $3.00 for software orders 4% shipping for hardware minimum $4.00 Master Card and Visa orders please include card # expiration date and signature WI residents please include 5% sales tax HI AK FPO APO Puerto Rico and Canadian orders please add 5% shipping minimum $5.00 All other foreign orders add 15% shipping minimum $10.00. All orders shipped outside the Continental U S A are shipped first class insured U S mail If foreign shipping charges exceed the minimum amount you will be charged the additional amount to get your package to you quickly and safely All goods are new and include factory warranty Due to our low prices all sales are final **All defective returns must have a return authorization number** Please call (414) 351-2007 to obtain an R A # or your return will not be accepted Priced and availability subject to change without notice

100 colors, prints text at 80 characters per second in standard mode, or 40 characters per second in a near-letter-quality mode. Pages widths are 80 columns in standard mode or 132 columns in condensed mode. Other modes include expanded, boldface, italic, fine print, underlining, superscripts, and subscripts. An Okimate 20 with Plug 'N Print Module retails for $268.
Okidata, 532 Fellowship Rd., Mt. Laurel, NJ 08054.

Circle Reader Service Number 229.

ST MIDI Software

Electronic Music Publishing House has announced new software to take advantage of the Atari ST's built-in MIDI (Musical Instrument Digital Interface) ports.

Midiplay turns an ST into a 16-channel digital player/recorder that gives you control over the music's tempo, key, and timbre. It can play prerecorded music through the computer or a MIDI-equipped synthesizer, record music from a MIDI synthesizer, and display the music on the screen as it plays. It can also play music in slow motion—as much as ten times slower without altering the pitch. Depending on available memory, up to 250,000 MIDI notes/events can be recorded, and more than 150,000 can be stored on a single-sided 3½-inch disk. *Midiplay* responds to MIDI START, MIDI STOP, and MIDI CONTINUE commands from a remote MIDI device, and it supports playback looping. Playback time is accurate to 1/1000 second.

The synthesizer section turns the ST into a velocity-sensitive, three-voice, realtime synthesizer with eight envelopes, envelope-release control, vibrato speed/depth controls, and storage/playback of up to 26 programmable sound patches.

Three prerecorded music disks will also be available: *Classics Volume I—Music of Bach, Beethoven, Chopin, Debussy, and Mozart; Classics Volume II—The Music of Amadeus Mozart;* and *Music of the Beatles.* Other music disks are planned for the future.

Midiplay will retail for $49.95. It requires only an Atari ST; a MIDI-equipped synthesizer is optional.
Electronic Music Publishing House, 2210 Wilshire Blvd., Suite 488, Santa Monica, CA 90403.

Circle Reader Service Number 230.

Print Utility For Atari ST

Unison World has introduced an Atari ST version of its bestselling *PrintMaster*, a do-it-yourself print shop that allows easy creation of cards, signs, calendars, banners, and stationery. The program includes over 100 high-resolution graphics and many predefined border designs, type fonts, styles, and layout patterns. Menu-driven operation makes the program very easy to use, even for someone with no programming or drawing skills.

Suggested retail price for the ST version is $39.95. Other versions available include IBM-PC ($59.95), Commodore 64 ($34.95), and Apple II ($39.95). Art Gallery disks with additional graphics are available at additional cost.
Unison World, 3165 Adeline St., Berkeley, CA 94703.

Circle Reader Service Number 231.

More Stickybear Software

Weekly Reader Software has added several new products to its line of educational software featuring the familiar character Stickybear. *Stickybear Math 2* helps children practice multiplication and division ($39.95). *Stickybear BASIC* is a gentle introduction to the BASIC programming language ($39.95). *Stickybear Printer* is a sophisticated, easy-to-use graphic design program ($39.95). And *Stickybear Car Builder* helps familiarize you with all the machanics of car building by letting you design, construct, refine, and test sample automobiles ($39.95).
Weekly Reader Family Software, 245 Long Hill Rd., Middletown, CT 06457.

Circle Reader Service Number 232.

New Casio Keyboards

Casio has introduced several new electronic keyboards. The Model MT-55 ($149.50) is a 44-key mini-keyboard with twelve instrument sounds, twelve auto-rhythms, and auto-chording. This six-note polyphonic instrument has a real-time memory that holds 512 melody notes or can be used to store auto-chording for ease of performance. The Model MT-205 ($199) is a 49-key stereo mini-keyboard with twelve instrument sounds. It features twelve auto-rhythms with intro, fill-in, and ending patterns. Optional DP-1 drum pads can be hooked up for manual play of the PCM drum sound sources. The unit is battery powered. The Model MT-88 ($199) is a 49-key mini-keyboard with twelve instrument sounds, twelve auto-rhythms, and auto-chording. It allows auto-play of songs stored in ROM packs. The keyboard's Chord Guide feature teaches the user to play 3-note fingered chords easily by following lights over the keyboard.
Casio, Inc., 15 Gardner Rd., Fairfield, NJ 07006.

Circle Reader Service Number 233.

Briefly Noted

New products of all kinds were introduced at June's Consumer Electronics Show in Chicago. Here are some highlights:

• **SSI** introduced its latest tactical Civil War game, *Gettysburg: The Turning Point,* for Apple II series, Commodore 64, Atari 400/800/1200, and IBM-PC at $59.95 each. *Strategic Simulations, 1046 N. Rengstorff Ave., Mountain View, CA 94043.*

• **Star Micronics** premiered an upgrade of the popular Gemini 10X printer, the Gemini II. It combines the best features from the earlier model with the ease-of-use found in office printers ($329). *Star Micronics, 200 Park Ave., Suite 3510, New York, NY 10166.*

• **Keypunch Software** is distributing a line of inexpensive entertainment, educational, and productivity software for IBM, Apple, Commodore, and Atari. Prices range from $6.99 to $9.99. *Keypunch Software, 1221 Pioneer Bldg., St. Paul, MN 55101.*

• **Main Street Publishing** offers a similar budget line of packages previously sold by other publishers. Prices range from $4.95 to $9.95. *Main Street Publishing, 611 W. Travelers Trail; Burnsville, MN 55337.*

• **Mastertronic's** latest releases include *Ninja, Elektraglide,* and *Video Poker.* For the Commodore 64 ($9.99). *Mastertronic International, 7311B Grove Rd., Frederick, MD 21701.*

• **BCI** introduced *Mind Over Matter,* which contains four self-help programs: "Lose Weight," "Stop Smoking," "Be Successful," and "Conquer Stress." For IBM-PC, Apple II, Commodore 64, and Atari 8-bit computers ($9.95 each). *BCI Software, P.O. Box 730, Ringwood, NJ 07456.*

• **First Star Software** premiered *Spy vs. Spy III: Arctic Antics* for Commodore 64 and 48K Atari ($29.95) and 64K Apple ($34.95). *First Star Software, 18 E. 41st St., New York, NY 10017.*

• **Intellicreations** (formerly Datasoft) introduced *Crosscheck,* a combination crossword puzzle/Scrabble game for Atari 8-bit, Commodore 64, Apple II ($29.95), and IBM-PC ($39.95). *Intellicreations, 19808 Nordhoff Pl., Chatsworth, CA 91311.*

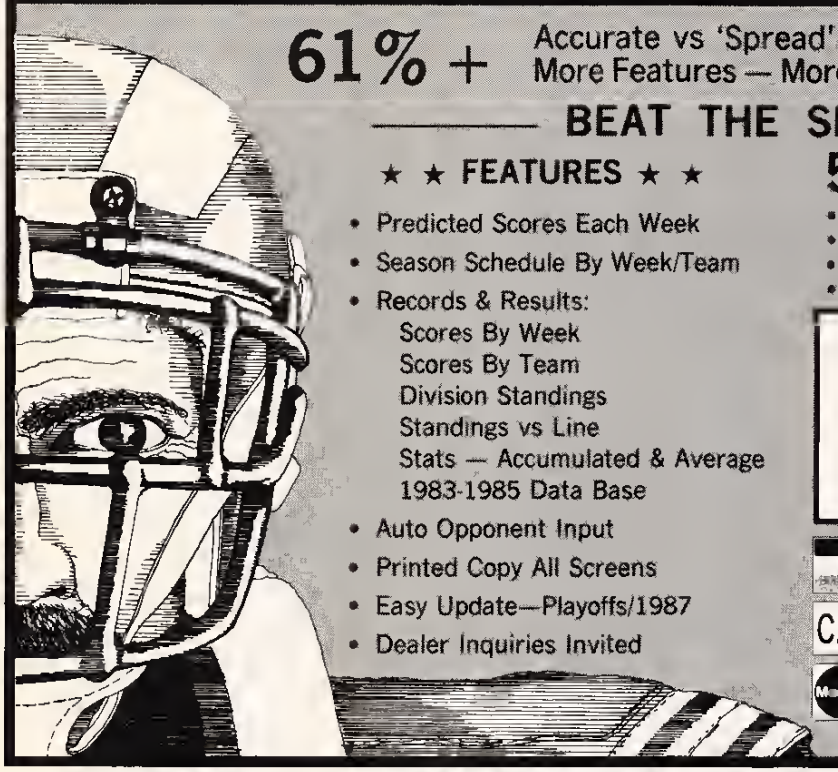
• **Sharedata** premiered the Home Companion series, a line of $9.95 programs geared toward home repair and maintenance. *Sharedata, 7122 Shady Oak Rd., Eden Prairie, MN 55344.*

©

PIK'EM 86

★ ★ ★ ★

A Complete Pro Football Prediction Program For The 1986 NFL SEASON



61% + Accurate vs 'Spread' Since 1981
More Features — More Information In 1986 To . . . . . .

BEAT THE SPREAD

★ ★ FEATURES ★ ★

- Predicted Scores Each Week
- Season Schedule By Week/Team
- Records & Results:
  - Scores By Week
  - Scores By Team
  - Division Standings
  - Standings vs Line
  - Stats — Accumulated & Average
  - 1983-1985 Data Base
- Auto Opponent Input
- Printed Copy All Screens
- Easy Update—Playoffs/1987
- Dealer Inquiries Invited

54.95 INCLUDES DISK, DOCUMENTATION SHIPPING & HANDLING

- Apple II — IIc — IIe
- Commodore 64 — 128
- IBM PC, Tandy & Compatibles
- TRS-80 MIII/IV

Stats Needed To Run Program Available In Local Newspapers
Or
We Will Furnish Stats By Mail Or Modem All 20 Weeks — Season Price . . . 40.00

Program Comes Updated Thru Current Week Of Season . . . No Extra Charge

VISA C.O.D. MasterCard

ORDERS SHIPPED IN 2 DAYS
PHONE LINES OPEN 24 HRS.

TOLL FREE — 800-722-2277
TEXAS COLL. — 214-586-8212

MARATHON SOFTWARE DEPT. C
P. O. BOX 1349
JACKSONVILLE, TEXAS 75766

TOLL FREE & COLLECT FOR ORDERS ONLY

$5 TALKING DISK

OVER 100 WORDS in vocabularies that you can put into your own programs! No extra hardware required. Sample programs include:

- Talking four-function calculator — choose English, Spanish, or German.
- Talking keyboard — letters and punctuation in English.
- Demonstration of voice editing.

The $5 Talking Disk is available for Commodore 64, 128, Atari 800, 800XL, 130XE, and Apple II+ (64K), IIe, and IIc.

If you want to make your own vocabularies, in any language or accent, then you must have a VOICE MASTER for processing speech input. VOICE MASTER lets you do much more: YOU CAN RECOGNIZE SPOKEN COMMANDS and WRITE MUSIC AS YOU HUM! And affordable too — under $90 including the headset and microphone.

Send $5 for the talking disk. Prepaid orders only. Includes postage. ($7 outside USA, Canada, and Mexico.) Information on VOICE MASTER will accompany your disk. Or you can call or write for VOICE MASTER information separately. Please specify computer make and model with your order.

COVOX INC.
675 Conger St., Dept. C!
Eugene, OR 97402
Telephone (503) 342-1271

PROF JONES
SUMMER/FALL 1986

ORDER LINE
(208) 342-5533

THOROUGHBRED
GREYHOUND
TROTTER
LOTTERY/LOTTO
FOOTBALL
BASKETBALL
OFFICE HOURS
M-F 8 AM - 5 PM
SAT 9 AM - 1 PM
(Mountain Standard Time)

FREE CATALOG

1940 W. STATE
BOISE, IDAHO 83702

VISA
AMEX

PROFESSIONAL HANDICAPPING SYSTEMS

PROFESSIONAL SERIES™ (Tho/Grey/Trot)
The all new Professional Series™ represents the most advanced handicapping software available

Analysis Module™
Complete bet analysis highlights this basic Professional Series™ module Full 50 tracks/kennels/etc $249.95

Factor Value/Multiple Regression Module™
Factor Value Weighting highlights this addition module™ $149.95

Data Base Manager Module™
Automatic storage of last 11 races highlights this module ($99.95 with Factor Value Module) $149.95

GOLD EDITION™ (Tho/Grey/Trot)
The classic Gold Edition™ from Prof Jones offers flexibility, results, and ease of use

Gold Edition™	$159.95
Enhanced Gold Edition™	$199.95
Limited Gold Edition™	$299.95
Ultra Edition™	$399.95

Professor Picks Football™
$99.95, with win/loss power ratings $149.95. Professional Series™ $199.95

Expanded Lottery/Lotto Analysis
Lottery 3-4 digits $79.95, Lotto max of 99 digits $99.95, Enhanced Lottery/ Lotto $129.95

PC-3 Portable Computer (4k)
Choice of Thoroughbred/Greyhound/Trotter Gold Edition™ software $249.95

Model 100 Portable Computer (32k)
Choice of Thoroughbred/Greyhound/Trotter Gold Edition™ software with Master Bettor™ $649.95

Handicapper's Bulletin Board now available
VHS Training Tapes now available

Terms: Free shipping all software Add $6.00 COD / $6.00 UPS Blue / $9.00 Out-of-country / ID residents add 5% / 3 weeks personal checks / cash price only add 2% Visa, MC, AMEX Prices subject to change

Atari Sound Development System

If you've tried to use the Sound Editor (Program 1, p. 71) from this article in the July issue, you've no doubt discovered that something is missing. The last 53 lines of the program were accidentally omitted. To complete the listing, add the following lines:

```
BC 2840 ? :? :? :POKE 752,0
CB 2850 ? "Enter name for LO
        AD file."
FN 2860 ? " or X to exit."
PB 2870 ? "{3 SPACES}{Q}{22 R}
        {E}"
OH 2880 ? "{3 SPACES}|On:fil
        ename.  Extender!"
AF 2890 ? "{3 SPACES}|automa
        tically attached!"
PH 2900 ? "{3 SPACES}{Z}{22 R}
        {C}"
MO 2910 GOSUB 2630:IF FL$="X
        " THEN RETURN
IB 2920 IF PEEK(195)<>170 TH
        EN 2940
AM 2930 ? :? FL$;" does not
        seem to exist...":PO
        KE 752,1:POSITION 0,
        20:? "{8 SPACES}PRES
        S ANY KEY":GET #1,K:
        GOTO 2830
IO 2940 ? "Okay, loading ";F
        L$;"."
MJ 2950 CLOSE #2:OPEN #2,4,0
        ,FL$:GET #2,8YTE
BM 2960 FOR X=0 TO 3:FOR Y=0
         TO 1:GET #2,Z:SO(X,
        Y)=Z:NEXT Y:NEXT X
DB 2970 FOR X=0 TO 3:FOR Y=1
         TO 35:GET #2,Z:S(X,
        Y)=Z:NEXT Y:NEXT X
DD 2980 8YT=8YTE
DC 2990 FOR X=7 TO 0 STEP -1
        :Y=INT(2^X+0.5):IF 8
        YT>=Y THEN 8YT=8YT-Y
        :8IT(X)=1
FK 3000 NEXT X
KE 3010 RETURN
NB 3020 REM INITIALIZATION
NB 3030 GRAPHICS 0:POKE 82,0
        :POKE 710,0:POKE 752
        ,1:DIM FLL$(20),FL$(
        20),8IT(7),VO(3),STA
        T(3):POKE 559,0
CO 3040 FOR J=0 TO 7:8IT(J)=
        0:NEXT J
GN 3050 FOR J=0 TO 3:VO(J)=8
        :STAT(J)=1:NEXT J
AA 3060 OPEN #1,4,0,"K:"
JN 3070 SOUND 0,50,10,10:FOR
         D=1 TO 5:NEXT D:SOU
        ND 0,0,0,0
OP 3080 DIM SO(3,2),S(3,35),
        G(15,4)
OJ 3090 RESTORE 3100:FOR X=0
         TO 3:FOR Y=0 TO 2:R
        EAD D:SO(X,Y)=D:NEXT
         Y:NEXT X
KB 3100 DATA 243,160,53760,1
        93,160,53762,144,160
        ,53764,121,160,53766
AF 3110 FOR X=0 TO 3:FOR Y=1
         TO 35:S(X,Y)=0:NEXT
         Y:NEXT X
MF 3120 SOUND 0,100,10,10:FO
        R D=1 TO 5:NEXT D:SO
        UND 0,0,0,0
NA 3130 RESTORE 3150:FOR X=1
         TO 15:FOR Y=1 TO 4:
        READ D:G(X,Y)=D:NEXT
         Y:NEXT X
MM 3140 SOUND 0,150,10,10:FO
        R D=1 TO 5:NEXT D:SO
        UND 0,0,0,0
IF 3150 DATA 1,1,1,1,1,1,1,0
        ,1,1,0,1,1,1,0,0,1,0
        ,1,1,1,0,1,0,1,0,0,1
        ,0,1,1,1,0,1,1,0,0,1
        ,0,1,0,0,1,1,1,0,0,0
        ,0,1,0,0
JK 3160 DATA 0,0,1,0,0,0,0,1
AL 3170 GRAPHICS 0:POKE 710,
        0:POKE 709,10:POKE 7
        52,1:POKE 559,0
KO 3180 OL=PEEK(560)+256*PEE
        K(561)
OP 3190 MEMTOP=PEEK(742)
HH 3200 SCREEN1=PEEK(89):SCR
        EEN2=MEMTOP-5
NL 3210 POKE 89,SCREEN2:POKE
         106,SCREEN2+4:POKE
        DL+5,SCREEN2:? CHR$(
        125)
CJ 3220 FOR D=5 TO 20:POSITI
        ON 3,D:? "!!!!!!!!!!
        !!!!!!!!!!!!!!!!!!!
        !!!!!":NEXT D
MI 3230 SOUND 0,200,10,10:FO
        R D=1 TO 5:NEXT D:SO
        UND 0,0,0,0
KN 3240 POSITION 1,0:? "ENVE
        LOPE EDITOR"
DL 3250 POSITION 2,1:? "for
        Voice #"
NM 3260 POSITION 2,2:? "Pitc
        h value: "
AI 3270 FOR N=15 TO 10 STEP
        -1:Y=20-N:X=0:POSITI
        ON X,Y:? N:NEXT N
KL 3280 FOR N=9 TO 0 STEP -1
        :Y=20-N:X=0:POSITION
         X,Y:? N:NEXT N
OJ 3290 POSITION 3,22:? "123
        4567890123456789012
        3456789012345"
BO 3300 POSITION 20,0:? "L -
         Listen"
EM 3310 POSITION 20,1:? "M -
         Menu"
AJ 3320 POSITION 20,2:? "S -
         Change Sound"
BL 3330 POSITION 20,3:? "C -
         Clear Bars"
NE 3340 SOUND 0,255,10,10:FO
        R D=1 TO 5:NEXT D:SO
        UND 0,0,0,0
HM 3350 POKE 89,SCREEN1:POKE
         106,SCREEN1+4:POKE
        DL+5,SCREEN1:? CHR$(
        125):POKE 559,34
BJ 3360 GOTO 40
```

Minding IBM Memory

The correction in last month's CAPUTE! column is not sufficient to correct all the bugs in the deallocation routine for this article from the June issue (p. 85). The **mov bx, [bp+6]** instructions in Program 2 should instead be **mov bx,[bp+8]**. To make this correction in Program 3, replace lines 100–110 with the following:

```
AH 100 DATA &h55,&h06,&h8b,&hec,
       &h8b,&h5e,&h08,&h8e,&h07,
       &hb4,&h49,&hcd
KF 110 DATA &h21,&h8b,&h5e,&h08,
       &h89,&h07,&h07,&h5d,&hca,
       &h02,&h00
```

The version of the program which appears on the COMPUTE! Disk for April–June includes all corrections.

Hex War For Amiga

The Amiga version (Program 7, p. 55) of this game in the July issue uses the lowercase letter l as a variable name in several places. Unfortunately, on the printer used to make the listing there is no distinction between l and the numeral 1, so it's difficult to tell when to use the letter and when to use the number. Here are the places where the character should be an l (for clarity, change these to uppercase L): In the lines following the ones with labels 710, 715, and 718, the expressions should be **L = cit(j,1)**, **x = (k−L)*2+19**, and **y = (12−(k+L))*2+3.** Following the DATA statements in the Strengths routine, there is a loop that should use **FOR L=1 TO 5** and **NEXT L**. In the Reveille routine, there is a loop that should use **FOR L=0 to 6**, **army(k,L,pn) = army(k+1,L,pn)**, **army(k+1,L,pn) = 0**, and **NEXT L**. In the Prisoners subroutine there is a loop that should use **FOR L=0 TO 6** and **army(k,L,j) = 0**. The lines labeled 3480 and 3490 should both start with **L=**, and just below those are two other statements that should read **IF c(1) => L THEN** and **IF c(2) => L THEN**. ©

Pigskin Predictions! Pro Handicapper

Tired of wrestling with Sunday point spreads? Let your IBM PC or Commodore 64/128 do it for you! Pigskin Predictions, the best-selling NFL handicapper, takes the hassle out of rating National Football League games. Forget about obscure, meaningless statistics. Just spend a few minutes typing in each week's scores and let our point spread software go to work. Here's what Pigskin Predictions has to offer:

- Predicts point spreads for all games--for the current week and the remainder of the season.
- Calculates projected won-lost records for all weeks.
- Menu-driven selection of schedules, ratings, division races, predictions or results by team or week. Seven different reports to screen or printer!
- Maintains home field advantage and power ratings for all NFL teams.
- 1986 Schedule data file included free. Yearly updates available.

Pigskin Predictions is only $39.95 on disk. Versions available for all Commodore 64/128 and IBM/Competible computers. Get your copy now and be ready for the season!

Use your computer to improve your performance at the track! Separate programs for **Thoroughbreds, harness** horses and **greyhounds** rank the horses or dogs in each race quickly and easily, even if you've never handicapped before!

All the information you need is readily available in the thoroughbred Racing Form, harness or dog track program. We even provide a chart showing you exactly where to get the information you need! Our software puts the savvy of a veteran handicapper at your fingertips. Our complete instructions and wagering guide tell you how to bet, which races to bet and which ones to avoid—one of the real secrets of winning at the track!

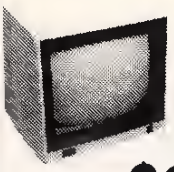
Thoroughbred factors include speed, distance, past performance, weight, class, jockey's record, beaten favorite and post position. **Harness factors** include speed, post position, driver's record, breaking tendencies, class, parked-out signs and beaten favorite. **Greyhound factors** include speed, past performance, maneuvering ability, favorite box, class, kennel record, beaten favorite and breaking ability.

Thoroughbred, harness and greyhound programs are sold separately. **IBM/Compatible** and **Apple II** versions are $49.95 each, any two for $74.95, all three $99.95. **Commodore 64/128** and **Tandy Color Computer** versions are $39.95 each on tape or disk. Any two for $59.95, all three $79.95.

Federal Hill Software
8134 Scotts Level Rd.
Baltimore, MD 21208
Orders 800-628-2828 Ext. 850
Information 301-521-4886

BELOW DEALER COST

MONITORS



FULL COLOR **$99**

RGB

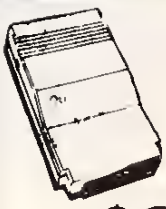
$259

MODEMS

300 BAUD

$28

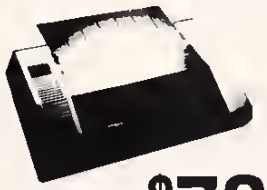
1200 BAUD



$88

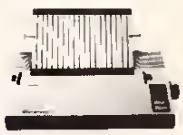
PRINTERS

EPSON



$78

NEAR LETTER QUALITY



$149

DISK DRIVE

1541 COMMODORE

$149

QUICK DELIVERY
1-800-345-5080

PRO-TECH-TRONICS
6870 Shingle Crk Pkwy
#103
Minneapolis MN 55430

From the publishers of *COMPUTE!*

August 1986 *COMPUTE!* Disk

All the exciting programs from the past three issues of *COMPUTE!* are on one timesaving, error-free, floppy disk that is ready to load on your Apple II, II+, IIe, and IIc computers. The August 1986 *COMPUTE! Disk* contains the entertaining and useful Apple programs from the June, July, and August 1986 issues of *COMPUTE!*.

The August 1986 *COMPUTE! Disk* costs $12.95 plus $2.00 shipping and handling and is available only from COMPUTE! Publications.

For added savings and convenience, you may also subscribe to the *COMPUTE! Disk*. At a cost of only $39.95 a year (a $12.00 savings), you'll receive four disks, one every three months. Each disk will contain all the programs for your machine from the previous three issues of *COMPUTE!*.

This is an excellent way to build your software library while you enjoy the quality programs from *COMPUTE!*.

Disks and subscriptions are available for Apple, Atari, Commodore 64 and 128, and IBM personal computers. Call for details.

For more information or to order the August 1986 *COMPUTE! Disk,* call toll free 1-800-346-6767 (in NY 212-265-8360) or write *COMPUTE! Disk*, P.O. Box 5038, F.D.R. Station, New York, NY 10150.

COMPUTE!™ Publications, Inc. abc
Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies
825 7th Avenue, 6th Floor, New York, NY 10019
Publishers of COMPUTE! COMPUTE!'s Gazette COMPUTE!'s Gazette Disk, COMPUTE! Books, and COMPUTE!'s Apple Applications

COMPUTE!'s Guide To Typing In Programs

Computers are precise—type the program *exactly* as listed, including necessary punctuation and symbols, except for special characters noted below. We have provided a special listing convention as well as a program to check your typing—"The Automatic Proofreader."

Programs for the IBM, TI-99/4A, and Atari ST models should be typed exactly as listed; no special characters are used. Programs for Commodore, Apple, and Atari 400/800/XL/XE computers may contain some hard-to-read special characters, so we have a listing system that indicates these control characters. You will find these Commodore and Atari characters in curly braces; *do not type the braces.* For example, {CLEAR} or {CLR} instructs you to insert the symbol which clears the screen on the Atari or Commodore machines. A complete list of these symbols is shown in the tables below. For Commodore, Apple, and Atari, a single symbol by itself within curly braces is usually a control key or graphics key. If you see {A}, hold down the CONTROL key and press A. This will produce a reverse video character on the Commodore (in quote mode), a graphics character on the Atari, and an invisible control character on the Apple.

Graphics characters entered with the Commodore logo key are enclosed in a special bracket: [<A>]. In this case, you would hold down the Commodore logo key as you type A. Our Commodore listings are in uppercase, so shifted symbols are underlined. A graphics heart symbol (SHIFT-S) would be listed as S. One exception is {SHIFT-SPACE}. When you see this, hold down SHIFT and press the space bar. If a number precedes a symbol, such as {5 RIGHT}, {6 S}, or [<8 Q>], you would enter five cursor rights, six shifted S's, or eight Commodore-Q's. On the Atari, inverse characters (white on black) should be entered with the inverse video

Atari 400/800/XL/XE

When you see	Type	See	
{CLEAR}	ESC SHIFT <		Clear Screen
{UP}	ESC CTRL -		Cursor Up
{DOWN}	ESC CTRL =		Cursor Down
{LEFT}	ESC CTRL +		Cursor Left
{RIGHT}	ESC CTRL *		Cursor Right
{BACK S}	ESC DELETE		Backspace
{DELETE}	ESC CTRL DELETE		Delete character
{INSERT}	ESC CTRL INSERT		Insert character
{DEL LINE}	ESC SHIFT DELETE		Delete line
{INS LINE}	ESC SHIFT INSERT		Insert line
{TAB}	ESC TAB		TAB key
{CLR TAB}	ESC CTRL TAB		Clear tab
{SET TAB}	ESC SHIFT TAB		Set tab stop
{BELL}	ESC CTRL 2		Ring buzzer
{ESC}	ESC ESC		ESCape key

Commodore PET/CBM/VIC/64/128/16/+4

When You Reed:	Press:	See:
{CLR}	SHIFT + CLR/HOME	
{HOME}	CLR/HOME	
{UP}	SHIFT + ↑ CRSR ↓	
{DOWN}	↑ CRSR ↓	
{LEFT}	SHIFT + ← CRSR →	
{RIGHT}	← CRSR →	
{RVS}	CTRL + 9	
{OFF}	CTRL + 0	
{BLK}	CTRL + 1	
{WHT}	CTRL + 2	
{RED}	CTRL + 3	
{CYN}	CTRL + 4	
{PUR}	CTRL + 5	
{GRN}	CTRL + 6	
{BLU}	CTRL + 7	
{YEL}	CTRL + 8	

When You Reed:	Press:	See:
[<1>]	COMMODORE + 1	
[<2>]	COMMODORE + 2	
[<3>]	COMMODORE + 3	
[<4>]	COMMODORE + 4	
[<5>]	COMMODORE + 5	
[<6>]	COMMODORE + 6	
[<7>]	COMMODORE + 7	
[<8>]	COMMODORE + 8	
{ F1 }	f1	
{ F2 }	SHIFT + f1	
{ F3 }	f3	
{ F4 }	SHIFT + f3	
{ F5 }	f5	
{ F6 }	SHIFT + f5	
{ F7 }	f7	
{ F8 }	SHIFT + f7	
←	←	

key (Atari logo key on 400/800 models).

Whenever more than two spaces appear in a row, they are listed in a special format. For example, {6 SPACES} means press the space bar six times. Our Commodore listings never leave a single space at the end of a line, instead moving it to the next printed line as {SPACE}.

Amiga program listings contain only one special character, the left arrow (←) symbol. This character marks the end of each program line. Wherever you see a left arrow, press RETURN or move the cursor off the line to enter that line into memory. Don't try to type in the left arrow symbol; it's there only as a marker to indicate where each program line ends.

The Automatic Proofreader

Type in the appropriate program listed below, then save it for future use. The Commodore Proofreader works on the Commodore 128, 64, Plus/4, 16, and VIC-20. Don't omit any lines, even if they contain unfamiliar commands or you think they don't apply to your computer. When you run the program, it installs a machine language program in memory and erases its BASIC portion automatically (so be sure to save several copies before running the program for the first time). If you're using a Commodore 128, Plus/4 or 16, do *not* use any GRAPHIC commands while the Proofreader is active. You should disable the Commodore Proofreader before running any other program. To do this, either turn the computer off and on or enter SYS 64738 (for the 64), SYS 65341 (128), SYS 64802 (VIC-20), or SYS 65526 (Plus/4 or 16). To reenable the Proofreader, reload the program and run it as usual. Unlike the original VIC/64 Proofreader, this version works the same with disk or tape.

On the Atari, run the Proofreader to activate it (the Proofreader remains active in memory as a machine language program); you must then enter NEW to erase the BASIC loader. Pressing SYSTEM RESET deactivates the Atari Proofreader; enter PRINT USR(1536) to reenable it.

The Apple Proofreader erases the BASIC portion of itself after you run it, leaving only the machine language portion in memory. It works with either DOS 3.3 or ProDOS. Disable the Apple Proofreader by pressing CTRL-RESET before running another BASIC program.

The IBM Proofreader is a BASIC program that simulates the IBM BASIC line editor, letting you enter, edit, list, save, and load programs that you type. Type RUN to activate. Be sure to leave Caps Lock on, except when typing lowercase characters.

Once the Proofreader is active, try typing in a line. As soon as you press RETURN, either a hexadecimal number (on the Apple) or a pair of letters (on the Commodore, Atari, or IBM) appears. The number or pair of letters is called a *checksum*.

Compare the value displayed on the screen by the Proofreader with the checksum printed in the program listing in the magazine. The checksum is given to the left of each line number. Just type in the program a line at a time (without the printed checksum), press RETURN or Enter, and compare the checksums. If they match, go on to the next line. If not, check your typing; you've made a mistake. Because of the checksum method used, do not type abbreviations, such as ? for PRINT. On the Atari and Apple Proofreaders, spaces are not counted as part of the checksum, so be sure you type the right number of spaces between quote marks. The Atari Proofreader does not check to see that you've typed the characters in the right order, so if characters are transposed, the checksum still matches the listing. The Commodore Proofreader catches transposition errors and ignores spaces unless they're enclosed in quotation marks. The IBM Proofreader detects errors in spacing and transposition.

IBM Proofreader Commands

Since the IBM Proofreader replaces the computer's normal BASIC line editor, it has to include many of the direct-mode IBM BASIC commands. The syntax is identical to IBM BASIC. Commands simulated are LIST, LLIST, NEW, FILES, SAVE, and LOAD. When listing your program, press any key (except Ctrl-Break) to stop the listing. If you enter NEW, the Proofreader prompts you to press Y to be especially sure you mean yes.

Two new commands are BASIC and CHECK. BASIC exits the Proofreader back to IBM BASIC, leaving the Proofreader in memory. CHECK works just like LIST, but shows the checksums along with the listing. After you have typed in a program, save it to disk. Then exit the Proofreader with the BASIC command, and load the program as usual (this replaces the Proofreader in memory). You can now run the program, but you may want to resave it to disk. This will shorten it on disk and make it load faster, but it can no longer be edited with the Proofreader. If you want to convert an existing BASIC program to Proofreader format, save it to disk with SAVE *"filename"*,A.

Program 1: Atari Proofreader

By Charles Brannon, Program Editor

```
100 GRAPHICS 0
110 FOR I=1536 TO 1700:REA
    D A:POKE I,A:CK=CK+A:N
    EXT I
120 IF CK<>19072 THEN ? "E
    rror in DATA Statement
    s.  Check Typing.":END

130 A=USR(1536)
140 ? :? "Automatic Proofr
    eader Now Activated."
150 END
160 DATA 104,160,0,185,26,
    3,201,69,240,7
170 DATA 200,200,192,34,20
    8,243,96,200,169,74
180 DATA 153,26,3,200,169,
    6,153,26,3,162
190 DATA 0,189,0,228,157,7
    4,6,232,224,16
200 DATA 208,245,169,93,14
    1,78,6,169,6,141
210 DATA 79,6,24,173,4,228
    ,105,1,141,95
220 DATA 6,173,5,228,105,0
    ,141,96,6,169
230 DATA 0,133,203,96,247,
    238,125,241,93,6
240 DATA 244,241,115,241,1
    24,241,76,205,238
250 DATA 0,0,0,0,0,32,62,2
    46,8,201
260 DATA 155,240,13,201,32
    ,240,7,72,24,101
270 DATA 203,133,203,104,4
    0,96,72,152,72,138
280 DATA 72,160,0,169,128,
    145,88,200,192,40
290 DATA 208,249,165,203,7
    4,74,74,74,24,105
300 DATA 161,160,3,145,88,
    165,203,41,15,24
310 DATA 105,161,200,145,8
    8,169,0,133,203,104
320 DATA 170,104,168,104,4
    0,96
```

Program 2: IBM Proofreader

By Charles Brannon, Program Editor

```
10 'Automatic Proofreader Vers
   ion 3.0 (Lines 205,206 adde
   d/190 deleted/470,490 chang
   ed from V2.0)
100 DIM L$(500),LNUM(500):COLO
    R 0,7,7:KEY OFF:CLS:MAX=0:
    LNUM(0)=65536!
110 ON ERROR GOTO 120:KEY 15,C
    HR$(4)+CHR$(70):ON KEY(15)
     GOSUB 640:KEY (15) ON:GOT
    O 130
120 RESUME 130
130 DEF SEG=&H40:W=PEEK(&H4A)
140 ON ERROR GOTO 650:PRINT:PR
    INT"Proofreader Ready."
150 LINE INPUT L$:Y=CSRLIN-INT
    (LEN(L$)/W)-1:LOCATE Y,1
160 DEF SEG=0:POKE 1050,30:POK
    E 1052,34:POKE 1054,0:POKE
     1055,79:POKE 1056,13:POKE
     1057,28:LINE INPUT L$:DEF
     SEG:IF L$="" THEN 150
170 IF LEFT$(L$,1)=" " THEN L$
    =MID$(L$,2):GOTO 170
```

```
1B0 IF VAL(LEFT$(L$,2))=0 AND
    MID$(L$,3,1)=" " THEN L$=M
    ID$(L$,4)
200 IF ASC(L$)>57 THEN 260 'no
     line number, therefore co
    mmand
205 BL=INSTR(L$," "):IF BL=0 T
    HEN BL$=L$:GOTO 206 ELSE B
    L$=LEFT$(L$,BL-1)
206 LNUM=VAL(BL$):TEXT$=MID$(L
    $,LEN(STR$(LNUM))+1)
210 IF TEXT$="" THEN GOSUB 540
    :IF LNUM=LNUM(P) THEN GOSU
    B 560:GOTO 150 ELSE 150
220 CKSUM=0:FOR I=1 TO LEN(L$)
    :CKSUM=(CKSUM+ASC(MID$(L$,
    I))*I) AND 255:NEXT:LOCATE
     Y,1:PRINT CHR$(65+CKSUM/1
    6)+CHR$(65+(CKSUM AND 15))
    +" "+L$
230 GOSUB 540:IF LNUM(P)=LNUM
    THEN L$(P)=TEXT$:GOTO 150
    'replace line
240 GOSUB 5B0:GOTO 150 'insert
     the line
260 TEXT$="":FOR I=1 TO LEN(L$
    ):A=ASC(MID$(L$,I)):TEXT$=
    TEXT$+CHR$(A+32*(A>96 AND
    A<123)):NEXT
270 DELIMITER=INSTR(TEXT$," ")
    :COMMAND$=TEXT$:ARG$="":IF
     DELIMITER THEN COMMAND$=L
    EFT$(TEXT$,DELIMITER-1):AR
    G$=MID$(TEXT$,DELIMITER+1)
     ELSE DELIMITER=INSTR(TEXT
    $,CHR$(34)):IF DELIMITER T
    HEN COMMAND$=LEFT$(TEXT$,D
    ELIMITER-1):ARG$=MID$(TEXT
    $,DELIMITER)
280 IF COMMAND$<>"LIST" THEN 4
    10
290 OPEN "scrn:" FOR OUTPUT AS
     #1
300 IF ARG$="" THEN FIRST=0:P=
    MAX-1:GOTO 340
310 DELIMITER=INSTR(ARG$,"-"):
    IF DELIMITER=0 THEN LNUM=V
    AL(ARG$):GOSUB 540:FIRST=P
    :GOTO 340
320 FIRST=VAL(LEFT$(ARG$,DELIM
    ITER)):LAST=VAL(MID$(ARG$,
    DELIMITER+1))
330 LNUM=FIRST:GOSUB 540:FIRST
    =P:LNUM=LAST:GOSUB 540:IF
    P=0 THEN P=MAX-1
340 FOR X=FIRST TO P:N$=MID$(S
    TR$(LNUM(X)),2)+" "
350 IF CKFLAG=0 THEN A$="":GOT
    O 370
360 CKSUM=0:A$=N$+L$(X):FOR I=
    1 TO LEN(A$):CKSUM=(CKSUM+
    ASC(MID$(A$,I))*I) AND 255
    :NEXT:A$=CHR$(65+CKSUM/16)
    +CHR$(65+(CKSUM AND 15))+"
    "
370 PRINT #1,A$+N$+L$(X)
380 IF INKEY$<>"" THEN X=P
390 NEXT :CLOSE #1:CKFLAG=0
400 GOTO 130
410 IF COMMAND$="LLIST" THEN O
    PEN "lpt1:" FOR OUTPUT AS
    #1:GOTO 300
420 IF COMMAND$="CHECK" THEN C
    KFLAG=1:GOTO 290
430 IF COMMAND$<>"SAVE" THEN 4
    50
440 GOSUB 600:OPEN ARG$ FOR OU
    TPUT AS #1:ARG$="":GOTO 30
    0
450 IF COMMAND$<>"LOAD" THEN 4
    90
460 GOSUB 600:OPEN ARG$ FOR IN
    PUT AS #1:MAX=0:P=0
470 WHILE NOT EOF(1):LINE INPU
    T #1,L$:BL=INSTR(L$," "):B
    L$=LEFT$(L$,BL-1):LNUM(P)=
    VAL(BL$):L$(P)=MID$(L$,LEN
    (STR$(VAL(BL$)))+1):P=P+1:
    WEND
4B0 MAX=P:CLOSE #1:GOTO 130
490 IF COMMAND$="NEW" THEN INP
    UT "Erase program - Are yo
    u sure";L$:IF LEFT$(L$,1)=
    "y" OR LEFT$(L$,1)="Y" THE
    N MAX=0:LNUM(0)=65536!:GOT
    O 130:ELSE 130
500 IF COMMAND$="BASIC" THEN C
    OLDR 7,0,0:ON ERROR GOTO 0
    :CLS:END
510 IF COMMAND$<>"FILES" THEN
    520
515 IF ARG$="" THEN ARG$="A:"
    ELSE SEL=1:GOSUB 600
517 FILES ARG$:GOTO 130
520 PRINT"Syntax error":GOTO 1
    30
540 P=0:WHILE LNUM>LNUM(P) AND
     P<MAX:P=P+1:WEND:RETURN
560 MAX=MAX-1:FOR X=P TO MAX:L
    NUM(X)=LNUM(X+1):L$(X)=L$(
    X+1):NEXT:RETURN
5B0 MAX=MAX+1:FOR X=MAX TO P+1
     STEP -1:LNUM(X)=LNUM(X-1)
    :L$(X)=L$(X-1):NEXT:L$(P)=
    TEXT$:LNUM(P)=LNUM:RETURN
600 IF LEFT$(ARG$,1)<>CHR$(34)
     THEN 520 ELSE ARG$=MID$(A
    RG$,2)
610 IF RIGHT$(ARG$,1)=CHR$(34)
     THEN ARG$=LEFT$(ARG$,LEN(
    ARG$)-1)
620 IF SEL=0 AND INSTR(ARG$,".
    ")=0 THEN ARG$=ARG$+".BAS"
630 SEL=0:RETURN
640 CLOSE #1:CKFLAG=0:PRINT"St
    opped.":RETURN 150
650 PRINT "Error #";ERR:RESUME
     150
```

Program 3: Commodore Proofreader

By Philip Nelson, Assistant Editor

```
10 VEC=PEEK(772)+256*PEEK(773)
   :LO=43:HI=44
20 PRINT "AUTOMATIC PROOFREADE
   R FOR ";:IF VEC=42364 THEN
   {SPACE}PRINT "C-64"
30 IF VEC=50556 THEN PRINT "VI
   C-20"
40 IF VEC=35158 THEN GRAPHIC C
   LR:PRINT "PLUS/4 & 16"
50 IF VEC=17165 THEN LO=45:HI=
   46:GRAPHIC CLR:PRINT"128"
60 SA=(PEEK(LO)+256*PEEK(HI))+
   6:ADR=SA
70 FOR J=0 TO 166:READ BYT:POK
   E ADR,BYT:ADR=ADR+1:CHK=CHK
   +BYT:NEXT
80 IF CHK<>20570 THEN PRINT "*
    ERROR* CHECK TYPING IN DATA
    STATEMENTS":END
90 FOR J=1 TO 5:READ RF,LF,HF:
   RS=SA+RF:HB=INT(RS/256):LB=
   RS-(256*HB)
100 CHK=CHK+RF+LF+HF:POKE SA+L
   F,LB:POKE SA+HF,HB:NEXT
110 IF CHK<>22054 THEN PRINT "
   *ERROR* RELOAD PROGRAM AND
   {SPACE}CHECK FINAL LINE":EN
   D
120 POKE SA+149,PEEK(772):POKE
    SA+150,PEEK(773)
130 IF VEC=17165 THEN POKE SA+
   14,22:POKE SA+18,23:POKESA+
   29,224:POKESA+139,224
140 PRINT CHR$(147);CHR$(17);"
   PROOFREADER ACTIVE":SYS SA
150 POKE HI,PEEK(HI)+1:POKE (P
   EEK(LO)+256*PEEK(HI))-1,0:N
   EW
160 DATA 120,169,73,141,4,3,16
   9,3,141,5,3
170 DATA 88,96,165,20,133,167,
   165,21,133,168,169
180 DATA 0,141,0,255,162,31,18
   1,199,157,227,3
190 DATA 202,16,248,169,19,32,
   210,255,169,18,32
200 DATA 210,255,160,0,132,180
   ,132,176,136,230,180
210 DATA 200,185,0,2,240,46,20
   1,34,208,8,72
220 DATA 165,176,73,255,133,17
   6,104,72,201,32,208
230 DATA 7,165,176,208,3,104,2
   08,226,104,166,180
240 DATA 24,165,167,121,0,2,13
   3,167,165,168,105
250 DATA 0,133,168,202,208,239
   ,240,202,165,167,69
260 DATA 168,72,41,15,168,185,
   211,3,32,210,255
270 DATA 104,74,74,74,74,168,1
   85,211,3,32,210
280 DATA 255,162,31,189,227,3,
   149,199,202,16,248
290 DATA 169,146,32,210,255,76
   ,86,137,65,66,67
300 DATA 68,69,70,71,72,74,75,
   77,80,81,82,83,88
310 DATA 13,2,7,167,31,32,151,
   116,117,151,128,129,167,136
   ,137
```

Program 4: Apple Proofreader

By Tim Victor, Editorial Programmer

```
10 C = 0: FOR I = 768 TO 768 +
   68: READ A:C = C + A: POKE I
   ,A: NEXT
20 IF C < > 7258 THEN PRINT "ER
   ROR IN PROOFREADER DATA STAT
   EMENTS": END
30 IF PEEK (190 * 256) < > 76 T
   HEN POKE 56,0: POKE 57,3: CA
   LL 1002: GOTO 50
40 PRINT CHR$ (4);"IN#A$300"
50 POKE 34,0: HOME : POKE 34,1:
    VTAB 2: PRINT "PROOFREADER
   INSTALLED"
60 NEW
100 DATA 216,32,27,253,201,141
110 DATA 208,60,138,72,169,0
120 DATA 72,189,255,1,201,160
130 DATA 240,8,104,10,125,255
140 DATA 1,105,0,72,202,208
150 DATA 238,104,170,41,15,9
160 DATA 48,201,58,144,2,233
170 DATA 57,141,1,4,138,74
180 DATA 74,74,74,41,15,9
190 DATA 48,201,58,144,2,233
200 DATA 57,141,0,4,104,170
210 DATA 169,141,96
```

©

COMPUTE's Author Guide

Most of the following suggestions serve to improve the speed and accuracy of publication. COMPUTE! is primarily interested in new and timely articles on the Commodore 64/128, Atari, Apple, IBM PC/PCjr, Amiga, and Atari ST. We are much more concerned with the content of an article than with its style, but articles should be clear and well-explained.

The guidelines below will permit your good ideas and programs to be more easily edited and published:

1. The upper left corner of the first page should contain your name, address, telephone number, and the date of submission.

2. The following information should appear in the upper right corner of the first page. If your article is specifically directed to one make of computer, please state the brand name and, if applicable, the BASIC or ROM or DOS version(s) involved. In addition, *please indicate the memory requirements of programs.*

3. The underlined title of the article should start about 2/3 of the way down the first page.

4. Following pages should be typed normally, except that in the upper right corner there should be an abbreviation of the title, your last name, and the page number. For example: Memory Map/Smith/2.

5. All lines within the text of the article must be double- or triple-spaced. A one-inch margin should be left at the right, left, top, and bottom of each page. No words should be divided at the ends of lines. And please do not justify. Leave the lines ragged.

6. Standard typing paper should be used (no erasable, onionskin, or other thin paper) and typing should be on one side of the paper only (upper- and lowercase).

7. Sheets should be attached together with a paper clip. Staples should not be used.

8. If you are submitting more than one article, send each one in a separate mailer with its own tape or disk.

9. Short programs (under 20 lines) can easily be included within the text. Longer programs should be separate listings. *It is essential that we have a copy of the program, recorded twice, on a tape or disk.* If your article was written with a word processor, we also appreciate a copy of the text file on the tape or disk. Please use high-quality 10 or 30 minute tapes with the program recorded on both sides. The tape or disk should be labeled with the author's name, the title of the article, and, if applicable, the BASIC/ROM/DOS version(s). Atari tapes should specify whether they are to be LOADed or ENTERed. We prefer to receive Apple programs on disk rather than tape. Tapes are fairly sturdy, but disks need to be enclosed within plastic or cardboard mailers (available at photography, stationery, or computer supply stores).

10. A good general rule is to spell out the numbers zero through ten in your article and write higher numbers as numerals (1024). The exceptions to this are: Figure 5, Table 3, TAB(4), etc. Within ordinary text, however, the zero through ten should appear as words, not numbers. Also, symbols and abbreviations should not be used within text: use "and" (not &), "reference" (not ref.), "through" (not thru).

11. For greater clarity, use all capitals when referring to keys (RETURN, TAB, ESC, SHIFT), BASIC words (LIST, RND, GOTO), and three languages (BASIC, APL, PILOT). Headlines and subheads should, however, be initial caps only, and emphasized words are not capitalized. If you wish to emphasize, underline the word and it will be italicized during typesetting.

12. Articles can be of any length—from a single-line routine to a multi-issue series. The average article is about four to eight double-spaced, typed pages.

13. If you want to include photographs, they should be either 5×7 black and white glossies or color slides.

14. We do not consider articles which are submitted simultaneously to other publishers. If you wish to send an article to another magazine for consideration, please do not submit it to us.

15. COMPUTE! pays between $70 and $800 for published articles. In general, the rate reflects the length and quality of the article. Payment is made upon acceptance. Following submission (Editorial Department, COMPUTE! Magazine, P.O. Box 5406, Greensboro, NC 27403) it will take from four to eight weeks for us to reply. If your work is accepted, you will be notified by a letter which will include a contract for you to sign and return. *Rejected manuscripts are returned to authors who enclose a self-addressed, stamped envelope.*

16. If your article is accepted and you have since made improvements to the program, please submit an entirely new tape or disk and a new copy of the article reflecting the update. We cannot easily make revisions to programs and articles. It is necessary that you send the revised version as if it were a new submission entirely, but be sure to indicate that your submission is a revised version by writing, "Revision" on the envelope and the article.

17. COMPUTE! does not accept unsolicited product reviews. If you are interested in serving on our panel of reviewers, contact the Review Coordinator for details.

MLX Machine Language Entry Program For Commodore 64

Ottis Cowper, Technical Editor

"MLX" is a labor-saving utility that allows will help you enter machine language program listings without error. MLX is required to enter all Commodore 64 machine language programs published in COMPUTE!.

Type in and save some copies of MLX (you'll want to use it to enter future ML programs from COMPUTE!, COMPUTE!'s GAZETTE, and COMPUTE! books). When you're ready to enter an ML program, load and run MLX. You'll be asked for a starting address and an ending address. These addresses should appear in the article accompanying the MLX-format program listing you're typing.

If you're unfamiliar with machine language, the addresses (and all other values you enter in MLX) may appear strange. Instead of the usual decimal numbers you're accustomed to, these numbers are in *hexadecimal*—a base 16 numbering system commonly used by ML programmers. Hexadecimal—hex for short—includes the numerals 0–9 and the letters A–F. But don't worry—even if you know nothing about ML or hex, you should have no trouble using MLX.

After you enter the starting and ending addresses, you'll be offered the option of clearing the workspace. The data you enter with MLX is kept in a special reserved area of memory; clearing this workspace fills the reserved area with zeros, which makes it easier to find where you left off typing if you enter the listing in several sessions. Choose this option if you're starting to enter a new listing. If you're continuing a listing that's partially typed from a previous session, there's no point in clearing the workspace, since the data you load in will fill the area with whatever values were in workspace memory at the time of the last Save.

At this point, functions menu will appear. If you're just starting to type in a program, pick the first option, ENTER DATA, by pressing the E key. You'll be asked for an address; type the four-digit number at the start of the first line of the program listing. If you've already typed in part of a program, be sure to load the partially completed program before you resume entry, then choose the ENTER DATA option and type the line number where you left off typing at the end of the previous session. In any case, make sure the address you enter corresponds to the address of a line in the listing. Otherwise, you'll be unable to enter the data correctly. If you pressed E by mistake, you can return to the command menu by pressing RETURN alone when asked for the address. (You can get back to the menu from most options by pressing RETURN with no other input.)

Entering A Listing

Once you're in Enter mode, MLX prints the address for each program line for you. You then type in all nine numbers on that line, beginning with the first two-digit number after the colon (:). Each line represents eight data bytes and a checksum. Although an MLX-format listing appears similar to the "hex dump" listings from a machine language monitor program, the extra checksum number on the end allows MLX to check your typing.

When you enter a line, MLX recalculates the checksum from the eight bytes and the address and compares this value to the number from the ninth column. If the values match, you'll hear a bell tone, the data will be added to the workspace area, and the prompt for the next line of data will appear. But if MLX detects a typing error, you'll hear a low buzz and see an error message. The line will then be redisplayed for editing.

Invalid Characters Banned

Only a few keys are active while you're entering data, so you may have to unlearn some habits. You *do not* type spaces between the columns; MLX automatically inserts these for you. You *do not* press RETURN after typing the last number in a line; MLX automatically enters and checks the line after you type the last digit.

Only the numerals 0–9 and the letters A–F can be typed in. If you press any other key (with some exceptions noted below), you'll hear a warning buzz. MLX checks for transposed characters. If you're supposed to type in A0 and instead enter 0A, MLX will catch your mistake. There is one error that can slip past MLX: Because of the checksum formula used, MLX won't notice if you accidentally type FF in place of 00, and vice versa. And there's a very slim chance that you could garble a line and still end up with a combination of characters that adds up to the proper checksum. However, these mistakes should not occur if you take reasonable care while entering data.

Editing Features

To correct typing mistakes before finishing a line, use the INST/DEL key to delete the character to the left of the cursor. (The cursor-left key also deletes.) If you mess up a line really badly, press CLR/HOME to start the line over. The RETURN key is also active, but only before any data is typed on a line. Pressing RETURN at this point returns you to the command menu. After you type a character of data, MLX disables RETURN until the cursor returns to the start of a line. Remember, you can press CLR/HOME to quickly get to a line number prompt.

More editing features are available when correcting lines in which MLX has detected an error. To make corrections in a line that MLX has redisplayed for editing, compare the line on the screen with the one printed in the listing, then move the cursor to the mistake and type the correct key. The cursor left and right keys provide the normal cursor controls. (The INST/DEL key now works as an alternative cursor-left key.) You cannot move left beyond the first character in the line. If you try to move beyond the rightmost character, you'll reenter the line. During editing, RETURN is active; pressing it tells MLX to recheck the line. You can press the CLR/HOME key to clear the entire line if you want to start from scratch, or if you want to get to a line number prompt to use RETURN to get back to the menu.

Display Data

The second menu choice, DISPLAY DATA, examines memory and shows the contents in the same format as the program listing (including the checksum). When you press D, MLX asks you for a starting address. Be sure that the starting address you give corresponds to a line number in the listing. Otherwise, the checksum display will be meaningless. MLX displays program lines until it reaches the end of the program, at which point the menu is redisplayed. You can pause the display by pressing the space bar. (MLX finishes printing the current line before halting.) Press space again to restart the display. To break out of the display and

get back to the menu before the ending address is reached, press RETURN.

Other Menu Options

Two more menu selections let you save programs and load them back into the computer. These are SAVE FILE and LOAD FILE; their operation is quite straightforward. When you press S or L, MLX asks you for the filename. You'll then be asked to press either D or T to select disk or tape.

You'll notice the disk drive starting and stopping several times during a load or save. Don't panic; this is normal behavior. MLX opens and reads from or writes to the file instead of using the usual LOAD and SAVE commands. Disk users should also note that the drive prefix 0: is automatically added to the filename (line 750), so this should *not* be included when entering the name. This also precludes the use of @ for Save-with-Replace, so remember to give each version you save a different name.

Remember that MLX saves the entire workspace area from the starting address to the ending address, so the save or load may take longer than you might expect if you've entered only a small amount of data from a long listing. When saving a partially completed listing, make sure to note the address where you stopped typing so you'll know where to resume entry when you reload.

MLX reports the standard disk or tape error messages if any problems are detected during the save or load. (Tape users should bear in mind that Commodore computers are never able to detect errors during a save to tape.) MLX also has three special load error messages: INCORRECT STARTING ADDRESS, which means the file you're trying to load does not have the starting address you specified when you ran MLX; LOAD ENDED AT *address*, which means the file you're trying to load ends before the ending address you specified when you started MLX; and TRUNCATED AT ENDING ADDRESS, which means the file you're trying to load extends beyond the ending address you specified when you started MLX. If you see one of these messages and feel certain that you've loaded the right file, exit and rerun MLX, being careful to enter the correct starting and ending addresses.

The QUIT menu option has the obvious effect—it stops MLX and enters BASIC. The RUN/STOP key is disabled, so the Q option lets you exit the program without turning off the computer. (Of course, RUN/STOP-RESTORE also gets you out.) You'll be asked for verification; press Y to exit to BASIC, or any other key to return to the menu. After quitting, you can type RUN again and reenter MLX without losing your data, as long as you don't use the clear workspace option.

The Finished Product

When you've finished typing all the data for an ML program and saved your work, you're ready to see the results. The instructions for loading and using the finished product vary from program to program. Some ML programs are designed to be loaded and run like BASIC programs, so all you need to type is LOAD *"filename"*,8 for disk or LOAD *"filename"* for tape, and then RUN. Such programs will usually have a starting address of 0801. Other programs must be reloaded to specific addresses with a command such as LOAD *"filename"*,8,1 for disk or LOAD *"filename"*,1,1 for tape, then started with a SYS to a particular memory address. The most common starting address for such programs is 49152, which corresponds to MLX address C000. In either case, you should always refer to the article which accompanies the ML listing for information on loading and running the program.

An Ounce Of Prevention

By the time you finish typing in the data for a long ML program, you may have several hours invested in the project. Don't take chances—use our "Automatic Proofreader" to type MLX, and then test your copy *thoroughly* before first using it to enter any significant amount of data. Make sure all the menu options work as they should. Enter fragments of the program starting at several different addresses, then use the Display option to verify that the data has been entered correctly. And be sure to test the Save and Load options several times to ensure that you can recall your work from disk or tape. Don't let a simple typing error in MLX cost you several nights of hard work.

MLX

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue of COMPUTE!

```
EK 100 POKE 56,50:CLR:DIM IN$,
       I,J,A,B,A$,B$,A(7),N$
DM 110 C4=48:C6=16:C7=7:Z2=2:Z
       4=254:Z5=255:Z6=256:Z7=
       127
CJ 120 FA=PEEK(45)+Z6*PEEK(46)
       :BS=PEEK(55)+Z6*PEEK(56
       ):H$="0123456789ABCDEF"
SB 130 R$=CHR$(13):L$="{LEFT}"
       :S$=" ":D$=CHR$(20):Z$=
       CHR$(0):T$="{13 RIGHT}"
CQ 140 SD=54272:FOR I=SD TO SD
       +23:POKE I,0:NEXT:POKE
       {SPACE}SD+24,15:POKE 78
       B,52
FC 150 PRINT"{CLR}"CHR$(142)CH
       R$(8):POKE 532B0,15:POK
       E 53281,15
EJ 160 PRINT T$" {RED}{RVS}
       {2 SPACES}[8 @]
       {2 SPACES}"SPC(28)"
       {2 SPACES}{OFF}{BLU} ML
       X II {RED}{RVS}
       {2 SPACES}"SPC(28)"
       {12 SPACES}{BLU}"
FR 170 PRINT"{3 DOWN}
       {3 SPACES}COMPUTE!'S MA
       CHINE LANGUAGE EDITOR
       {3 DOWN}"
JB 180 PRINT"{BLK}STARTING ADD
       RESS[4]";:GOSUB300:SA=A
       D:GOSUB1040:IF F THEN1B
       0
GF 190 PRINT"{BLK}{2 SPACES}EN
       DING ADDRESS[4]";:GOSUB
       300:EA=AD:GOSUB1030:IF
       {SPACE}F THEN190
KR 200 INPUT"{3 DOWN}{BLK}CLEA
       R WORKSPACE [Y/N][4]";A
       $:IF LEFT$(A$,1)<>"Y"TH
       EN220
PG 210 PRINT"{2 DOWN}{BLU}WORK
       ING...";:FORI=BS TO BS+
       EA-SA+7:POKE I,0:NEXT:P
       RINT"DONE"
DR 220 PRINTTAB(10)"{2 DOWN}
       {BLK}{RVS} MLX COMMAND
       {SPACE}MENU {DOWN}[4]":
       PRINT T$"{RVS}E{OFF}NTE
       R DATA"
BD 230 PRINT T$"{RVS}D{OFF}ISP
       LAY DATA":PRINT T$"
       {RVS}L{OFF}OAD FILE"
JS 240 PRINT T$"{RVS}S{OFF}AVE
        FILE":PRINT T$"{RVS}Q
       {OFF}UIT{2 DOWN}{BLK}"
JH 250 GET A$:IF A$=N$ THEN250
HK 260 A=0:FOR I=1 TO 5:IF A$=
       MID$("EDLSQ",I,1)THEN A
       =I:I=5
FD 270 NEXT:ON A GOTO420,610,6
       90,700,280:GOSUB1060:GO
       TO250
EJ 280 PRINT"{RVS} QUIT ":INPU
       T"{DOWN}[4]ARE YOU SURE
        [Y/N]";A$:IF LEFT$(A$,
       1)<>"Y"THEN220
EM 290 POKE SD+24,0:END
JX 300 IN$=N$:AD=0:INPUTIN$:IF
       LEN(IN$)<>4THENRETURN
KF 310 B$=IN$:GOSUB320:AD=A:B$
       =MID$(IN$,3):GOSUB320:A
       D=AD*256+A:RETURN
PP 320 A=0:FOR J=1 TO 2:A$=MID
       $(B$,J,1):B=ASC(A$)-C4+
       (A$>"@")*C7:A=A*C6+B
JA 330 IF B<0 OR B>15 THEN AD=
       0:A=-1:J=2
GX 340 NEXT:RETURN
CH 350 B=INT(A/C6):PRINT MID$(
       H$,B+1,1);:B=A-B*C6:PRI
       NT MID$(H$,B+1,1);:RETU
       RN
RR 360 A=INT(AD/Z6):GOSUB350:A
       =AD-A*Z6:GOSUB350:PRINT
       ":";
BE 370 CK=INT(AD/Z6):CK=AD-Z4*
       CK+Z5*(CK>Z7):GOTO390
PX 380 CK=CK*Z2+Z5*(CK>Z7)+A
JC 390 CK=CK+Z5*(CK>Z5):RETURN
QS 400 PRINT"{DOWN}STARTING AT
       [4]";:GOSUB300:IF IN$<>
       N$ THEN GOSUB1030:IF F
       {SPACE}THEN400
EX 410 RETURN
HD 420 PRINT"{RVS} ENTER DATA
       {SPACE}":GOSUB400:IF IN
       $=N$ THEN220
```

```
JK 430 OPEN3,3:PRINT
SK 440 POKE198,0:GOSUB360:IF F
       THEN PRINT IN$:PRINT"
       {UP}{5 RIGHT}";
GC 450 FOR I=0 TO 24 STEP 3:B$
       =S$:FOR J=1 TO 2:IF F T
       HEN B$=MID$(IN$,I+J,1)
HA 460 PRINT"{RVS}"B$L$;:IF I<
       24THEN PRINT"{OFF}";
HD 470 GET A$:IF A$=N$ THEN470
FK 480 IF(A$>"/"ANDA$<":")OR(A
       $>"@"ANDA$<"G")THEN540
MP 490 IF A$=R$ AND((I=0)AND(J
       =1)OR F)THEN PRINT B$;:
       J=2:NEXT:I=24:GOTO550
KC 500 IF A$="{HOME}" THEN PRI
       NT B$:J=2:NEXT:I=24:NEX
       T:F=0:GOTO440
MX 510 IF(A$="{RIGHT}")ANDF TH
       ENPRINT B$L$;:GOTO540
GK 520 IF A$<>L$ AND A$<>D$ OR
       ((I=0)AND(J=1))THEN GOS
       UB1060:GOTO470
HG 530 A$=L$+S$+L$:PRINT B$L$;
       :J=2-J:IF J THEN PRINT
       {SPACE}L$;:I=I-3
QS 540 PRINT A$;:NEXT J:PRINT
       {SPACE}S$;
PM 550 NEXT I:PRINT:PRINT"{UP}
       {5 RIGHT}";:INPUT#3,IN$
       :IF IN$=N$ THEN CLOSE3:
       GOTO220
QC 560 FOR I=1 TO 25 STEP3:B$=
       MID${IN$,I):GOSUB320:IF
        I<25 THEN GOSUB380:A(I
       /3)=A
PK 570 NEXT:IF A<>CK THEN GOSU
       B1060:PRINT"{BLK}{RVS}
       {SPACE}ERROR: REENTER L
       INE [4]":F=1:GOTO440
HJ 580 GOSUB1080:B=BS+AD-SA:FO
       R I=0 TO 7:POKE B+I,A(I
       ):NEXT
QQ 590 AD=AD+8:IF AD>EA THEN C
       LOSE3:PRINT"{DOWN}{BLU}
       ** END OF ENTRY **{BLK}
       {2 DOWN}":GOTO700
GQ 600 F=0:GOTO440
QA 610 PRINT"{CLR}{DOWN}{RVS}
       {SPACE}DISPLAY DATA ":G
       OSUB400:IF IN$=N$ THEN2
       20
RJ 620 PRINT"{DOWN}{BLU}PRESS:
        {RVS}SPACE{OFF} TO PAU
       SE, {RVS}RETURN{OFF} TO
        BREAK[4]{DOWN}"
KS 630 GOSUB360:B=BS+AD-SA:FOR
       I=BTO B+7:A=PEEK(I):GOS
       UB350:GOSUB380:PRINT S$
       ;
CC 640 NEXT:PRINT"{RVS}";:A=CK
       :GOSUB350:PRINT
KH 650 F=1:AD=AD+8:IF AD>EA TH
       ENPRINT"{DOWN}{BLU}** E
       ND OF DATA **":GOTO220
KC 660 GET A$:IF A$=R$ THEN GO
       SUB1080:GOTO220
EQ 670 IF A$=S$ THEN F=F+1:GOS
       UB1080
AD 680 ONFGOTO630,660,630
CM 690 PRINT"{DOWN}{RVS} LOAD
       {SPACE}DATA ":OP=1:GOTO
       710
PC 700 PRINT"{DOWN}{RVS} SAVE
       {SPACE}FILE ":OP=0
RX 710 IN$=N$:INPUT"{DOWN}FILE
       NAME[4]";IN$:IF IN$=N$
       {SPACE}THEN220
PR 720 F=0:PRINT"{DOWN}{BLK}
       {RVS}T{OFF}APE OR {RVS}
       D{OFF}ISK: [4]";
FP 730 GET A$:IF A$="T"THEN PR
       INT"T{DOWN}":GOTO880
HQ 740 IF A$<>"D"THEN730
HH 750 PRINT"D{DOWN}":OPEN15,8
       ,15,"I0:":B=EA-SA:IN$="
       0:"+IN$:IF OP THEN810
SQ 760 OPEN 1,8,8,IN$+",P,W":G
       OSUB860:IF A THEN220
FJ 770 AH=INT(SA/256):AL=SA-(A
       H*256):PRINT#1,CHR$(AL)
       ;CHR$(AH);
PE 780 FOR I=0 TO B:PRINT#1,CH
       R$(PEEK(BS+I));:IF ST T
       HEN800
FC 790 NEXT:CLOSE1:CLOSE15:GOT
       O940
GS 800 GOSUB1060:PRINT"{DOWN}
       {BLK}ERROR DURING SAVE:
       [4]":GOSUB860:GOTO220
MA 810 OPEN 1,8,8,IN$+",P,R":G
       OSUB860:IF A THEN220
GE 820 GET#1,A$,B$:AD=ASC(A$+Z
       $)+256*ASC(B$+Z$):IF AD
       <>SA THEN F=1:GOTO850
RX 830 FOR I=0 TO B:GET#1,A$:P
       OKE BS+I,ASC(A$+Z$):IF(
       I<>B)AND ST THEN F=2:AD
       =I:I=B
FA 840 NEXT:IF ST<>64 THEN F=3
FO 850 CLOSE1:CLOSE15:ON ABS(F
       >0)+1 GOTO960,970
SA 860 INPUT#15,A,A$:IF A THEN
        CLOSE1:CLOSE15:GOSUB10
       60:PRINT"{RVS}ERROR: "A
       $
GQ 870 RETURN
EJ 880 POKE183,PEEK(FA+2):POKE
       187,PEEK(FA+3):POKE188,
       PEEK(FA+4):IFOP=0THEN92
       0
HJ 890 SYS 63466:IF(PEEK(783)A
       ND1)THEN GOSUB1060:PRIN
       T"{DOWN}{RVS} FILE NOT
       {SPACE}FOUND ":GOTO690
CS 900 AD=PEEK(829)+256*PEEK(8
       30):IF AD<>SA THEN F=1:
       GOTO970
SC 910 A=PEEK(831)+256*PEEK(83
       2)-1:F=F-2*(A<EA)-3*(A>
       EA):AD=A-AD:GOTO930
KM 920 A=SA:B=EA+1:GOSUB1010:P
       OKE780,3:SYS 63338
JF 930 A=BS:B=BS+(EA-SA)+1:GOS
       UB1010:ON OP GOTO950:SY
       S 63591
AE 940 GOSUB1080:PRINT"{BLU}**
        SAVE COMPLETED **":GOT
       O220
XP 950 POKE147,0:SYS 63562:IF
       {SPACE}ST>0 THEN970
FR 960 GOSUB1080:PRINT"{BLU}**
        LOAD COMPLETED **":GOT
       O220
DP 970 GOSUB1060:PRINT"{BLK}
       {RVS}ERROR DURING LOAD:
       {DOWN}[4]":ON F GOSUB98
       0,990,1000:GOTO220
PP 980 PRINT"INCORRECT STARTIN
       G ADDRESS (";:GOSUB360:
       PRINT")":RETURN
GR 990 PRINT"LOAD ENDED AT ";:
       AD=SA+AD:GOSUB360:PRINT
        D$:RETURN
FD 1000 PRINT"TRUNCATED AT END
       ING ADDRESS":RETURN
RX 1010 AH=INT(A/256):AL=A-(AH
       *256):POKE193,AL:POKE1
       94,AH
FF 1020 AH=INT(B/256):AL=B-(AH
       *256):POKE174,AL:POKE1
       75,AH:RETURN
FX 1030 IF AD<SA OR AD>EA THEN
       1050
HA 1040 IF(AD>511 AND AD<40960
       )OR(AD>49151 AND AD<53
       248)THEN GOSUB1080:F=0
       :RETURN
HC 1050 GOSUB1060:PRINT"{RVS}
       {SPACE}INVALID ADDRESS
        {DOWN}{BLK}":F=1:RETU
       RN
AR 1060 POKE SD+5,31:POKE SD+6
       ,208:POKE SD,240:POKE
       {SPACE}SD+1,4:POKE SD+
       4,33
DX 1070 FOR S=1 TO 100:NEXT:GO
       TO1090
PF 1080 POKE SD+5,8:POKE SD+6,
       240:POKE SD,0:POKE SD+
       1,90:POKE SD+4,17
AC 1090 FOR S=1 TO 100:NEXT:PO
       KE SD+4,0:POKE SD,0:PO
       KE SD+1,0:RETURN
```

©

COMPUTE!
TOLL FREE
Subscription
Order Line
1-800-247-5470
In IA 1-800-532-1272

COMPUTE! Subscriber Services

Please help us serve you better. If you need to contact us for any of the reasons listed below, write to us at:

COMPUTE! Magazine
P.O. Box 10954
Des Moines, IA 50340

or call the Toll Free number listed below.

Change Of Address. Please allow us 6–8 weeks to effect the change, send your current mailing label along with your new address.

Renewal. Should you wish to renew your **COMPUTE!** subscription before we remind you to, send your current mailing label with payment or charge number or call the Toll Free number listed below.

New Subscription. A one year (12 month) US subscription to **COMPUTE!** is $24.00 (2 years, $45.00, 3 years, $65.00. For subscription rates outside the US, see staff page). Send us your name and address or call the Toll Free number listed below.

Delivery Problems. If you receive duplicate issues of **COMPUTE!**, if you experience late delivery or if you have problems with your subscription, please call the Toll Free number listed below.

COMPUTE!
1-800-247-5470
In IA 1-800-532-1272

Classified

SOFTWARE

TANDY 1000 PROGRAMS AND NEWSLETTER Send for free information on educational & entertainment programs & newsletter. Soda Pop Software, POB 653, Kenosha, WI 53141

FREE SOFTWARE for C64, C128, IBM, & CPM Send SASE for info (specify computer) to: PUBLIC DOMAIN USERS GROUP PO Box 1442-A1, Orange Park, FL 32067

COMMODORE: TRY BEFORE YOU BUY. Top 25 best-selling games, utilities, new releases. Visa, MasterCard. Free brochure. Rent-A-Disk, 908 9th Ave., Huntington, WV 25701 (304) 522-1665

DISCOUNT SOFTWARE: Amiga/Apple/Atari/ C64-128/IBM PC-PCjr/TRS-80/Timex/Sinclair. Free Catalog. WMJ DATA SYSTEMS, 4 Butterfly Dr., Hauppauge, NY 11788

Genealogy for the 64/128. **Family Tree** produces Pedigree Charts, Fam. Group Records, Individ. Files, Indexes, Searches of Ancestors. LDS vers. avail.-$49.95. Genealogy Softw., POB 1151, Port Huron, MI 48061, (519) 344-3990

BRIDGE GAME PROGRAM $39.95 demo disk $5. 1 to 4 players for IBM, Apple, TI99, TRS80, C64, 128, +4, 16, VIC. Authors John & Lynda Allan, Box 313, Azilda, Ontario, Canada P0M 1B0

COMPUTER CASINO! 24 GAMES! BLACKJACK, 21, DRAW POKER, SLOTS, more! C64/128 Disk. $12.95. M. E. ADAMS, 6547 N. Acad. #446-A2, Co. Springs, CO 80907

AMIGA OR ATARI ST SOFTWARE. 20 Different disks $99, 10 disks $59, or $6.95 each. 30 C64 programs + catalog $2. Info-SASE FRUGALWARE, 23 E. Green St., W. Hazelton, PA 18201

Amiga and C64 Public Domain Software. For a list of available programs send a self-addressed, stamped envelope to MCA, P.O. Box 5533, Katy, TX 774491-5533

FREE APPLE SOFTWARE
Over 1000 Public Domain programs on 50 diskettes, $5 ea. + $1 s/h per order. Send $1 for catalog (refundable). C&H Enterprizes, Box 29243, Memphis, TN 38127

FREE SOFTWARE CATALOG!
Call Toll-Free 1-800-554-1162, Tevex, Inc. Save 1/3 off retail prices. We carry SSI, Elect. Arts, Infocom and many more!

SELL YOUR PROGRAMS to Software Publishers. Software for the Apple, Atari, Commodore, IBM, TI, TRS-80, Zenith, Osborne, Kaypro, Others. Directory $5.95. I.W., Box 40581, Pas., CA 91104

ATARI 8 BIT: FULL FUNCTION BUSINESS DBMS. Any drive(s)/upgrades. GL/AR/AP/Inv/ Mail/W/P. 9000 records/disk. MICROMOD, 1635-A Holden, Orlando, FL 32809 (305) 857-6014

FREE EVERYTHING BOOKS for TI99/4A and C64/C128! Our "Evarything Books" are packed full of the newest software, hardware, books, accessories and much more! The prices are low, the service is dependable, and you can order toll-free. Request your "Evarything Book" today. Call 1-800-348-2778 (219-259-7051 in Indiana) or write: TENEX Computer Express, P.O. Box 6578, South Bend, Indiana 46660. (K1A)

MISCELLANEOUS

IBM PCjr REPORT: THE NATIONAL NEWSLETTER PCjr-specific articles, reviews, Public Domain from across the nation. $18/yr. PCjr CLUB, POB 95067, Schaumburg, IL 60195

The Bard's Tale Hint book, maps, magic items, uses, for Apple or C64 for $12.50 Ck or MO. Send to Don Dannelley, 2914 Pennsylvania, Wichita Falls, TX 76309

COMPUTE! Classified is a low-cost way to tell over 350,000 microcomputer owners about your product or service.

Rates: $2S per line, minimum of four lines. Any or all of the first line set in capital letters at no charge. Add $15 per line for boldface words, or $50 for the entire ad set in boldface (any number of lines.) Inquire about display rates.

Terms: Prepayment is required. Check, money order, American Express, Visa, or MasterCard is accepted. Make checks payable to COMPUTE! Publications.

Form: Ads are subject to publisher's approval and must be either typed or legibly printed. One line equals 40 letters and spaces between words. Please underline words to be set in boldface.

General Information: Advertisers using post office box numbers in their ads must supply permanent address and telephone numbers. Ad will appear in next available issue after receipt.

Closing: 20th of the third month preceding cover date (e.g., June issue closes March 20th). Send order and remittance to: Harry Blair, Classified Manager, COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. To place an ad by phone, call Harry Blair at (919) 27S-9809.

Notice: COMPUTE! Publications cannot be responsible for offers or claims of advertisers, but will attempt to screen out misleading or questionable copy.

Advertisers Index

WE TOP APPLE AND COMMODORE BEAUTIFULLY

Thomson offers a whole new world of graphics capabilities for your Commodore or Apple IIC or IIE computers. And Thomson offers you more than the competition.

Thomson monitors offer these advantages:

- Compatible with IBM,® Apple,® Commodore,® Atari™ and others
- The choice of monochrome or color monitors with TV-grade to high resolution text and graphics
- Monochrome text-editing mode for color monitors
- Alternate use as cable or VCR monitor
- Broad range of the features you need at the prices you want

Because of its versatility and compatibility, you can still use your Thomson monitor if you switch computers; you'll never outgrow the capabilities of a Thomson monitor.

Who is Thomson? Thomson is a six billion dollar multi-national company. Unlike our competition, we design and manufacture our own monitors; so the quality you get is constant and superb.

Your Apple or Commodore computer is a great body. Choose a Thomson monitor, and give it a beautiful face.

For the name of the Thomson dealer nearest you, call 1-800-325-0464. In California call 1-213-568-1002. (Monday-Friday, 9 a.m. to 5:00 p.m. PST)

THOMSON

A SIGHT FOR SORE EYES.™

© 1986 Thomson Consumer Products Corporation 5731 W. Slauson Avenue, Suite 111, Culver City, CA 90230

Thomson is a trademark of Thomson S.A. Apple is a registered trademark of Apple Computer, Inc. Commodore is a registered trademark of Commodore Electronics Limited. Atari is a trademark of Atari, Inc. IBM is a registered trademark of International Business Machines Corp.

From Origin comes the long-awaited sequel to the award-winning Ultima™ III

Ultima IV

Quest of the Avatar

Available on Apple®

A state-of-the-art fantasy role-playing game of unprecedented magnitude by Lord British™.

Prepare yourself for a grand adventure: Ultima™ IV, sixteen times larger than Ultima III, is a milestone in computer gaming—one that challenges your physical and mental skills while testing the true fabric of your character.

Enter Britannia, kingdom of Lord British. Journey through terrain of infinite proportions, conversing with characters on hundreds of topics. Unravel the mysteries of a superior magic system. At each turn beware of daemons, dragons and long-dead wizards haunting the most tranquil of places. Encounters with parties of mixed enemy types test your strategic abilities. Shrewd use of terrain can lead to victory against seemingly impossible odds.

Survive this multi-quest fantasy, then begin the final conflict, your quest of the Avatar. The ultimate challenge—the self—awaits....

ORIGIN SYSTEMS INC. 340 HARVEY ROAD, MANCHESTER, NH 03103 (603) 644-3360

ULTIMA™ III sends you on an incredible fantasy role-playing journey through monster-plagued Sosaria in search of the elusive Exodus.

MOEBIUS™ takes you through the elemental planes of a colorful Oriental world of fantasy and adventure in search of the Orb of Celestial Harmony.

AUTODUEL™ is a futuristic, fast-paced strategy role-playing game where the right of way goes to the biggest guns.

OGRE™ is a strategy game fought on the nuclear battlefield of tomorrow as an inhuman juggernaut Cybertank battles conventional forces.

Ultima and Lord British are trademarks of Richard Garriott/Moebius is a trademark of Greg Malone/AutoDuel and Ogre are trademarks of Steve Jackson/Apple is a trademark of Apple Computer Inc/Previous Ultimas are not needed to enjoy Ultima IV **Authors wanted. Call us today.**

RETROMAGS

Our goal is to preserve classic video game magazines so that they are not lost permanently.

People interested in helping out in any capacity, please visit us at www.retromags.com.

No profit is made from these scans, nor do we offer anything available from the publishers themselves.

If you come across anyone selling releases from this site, please do not support them and do let us know.

Thank you!